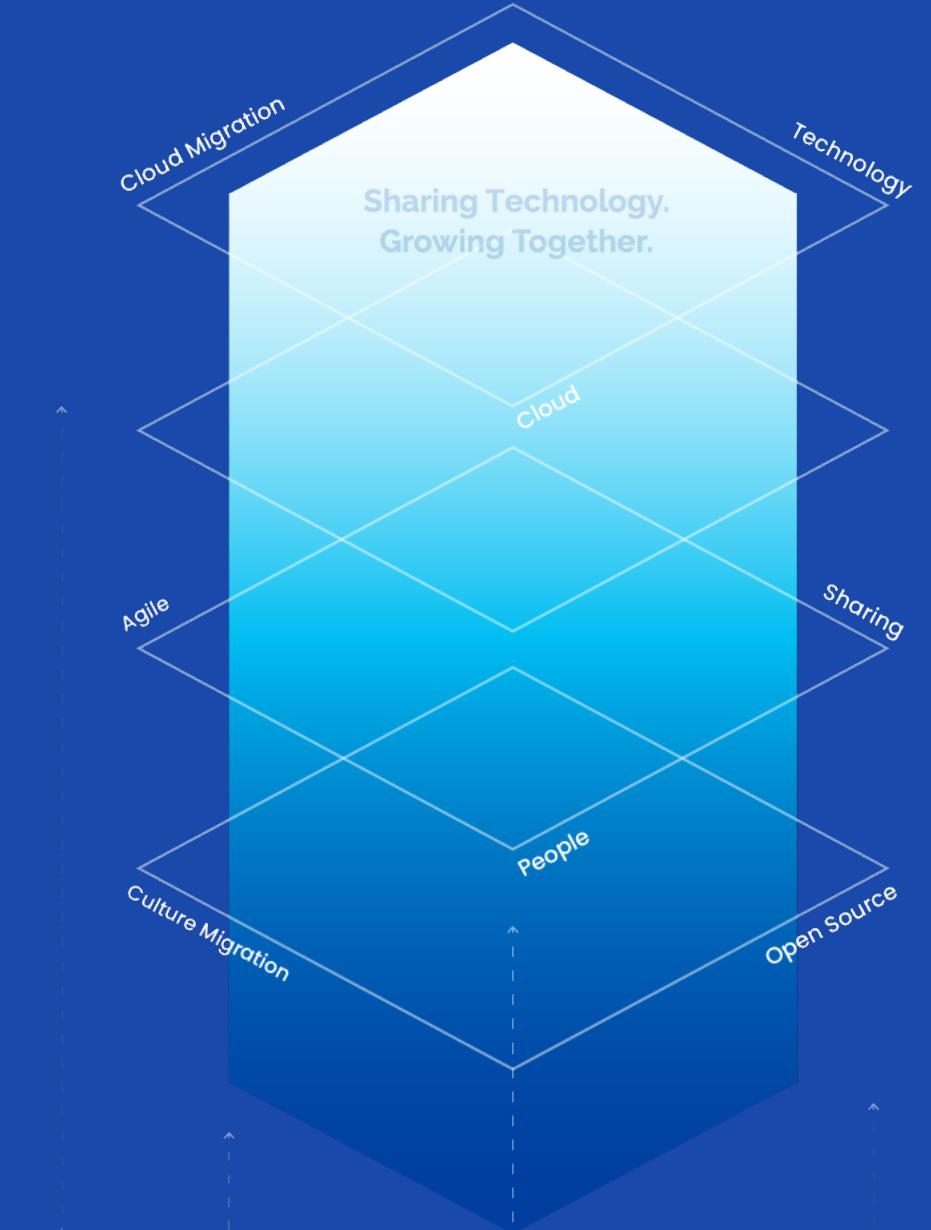


**Open Source Consulting Inc.**

# PaaS 서비스 패키지 구성방법과 실제 인프라 구현사례

2022.11.1 오픈소스컨설팅 김호진



# Contents

- 01 재단의 등장 – 오픈소스 패러다임의 변화**
- 02 필요한 오픈소스 선택하기**
- 03 필요한 오픈소스 통합 패키징 방법 – Helm**
- 04 오픈소스컨설팅 패키징 개요**
- 05 실제 인프라 구현 사례**

# 01

## 재단의 등장

- 오픈소스 패러다임의 변화



# 오픈소스의 글로벌 거버넌스로서의 재단의 역할



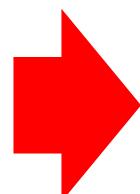
오픈소스의 안정성과 신뢰성의 패러다임을 바꾼 재단들의 등장 → OIF(구 Openstack Foundation) & CNCF (Cloud Native Computing Foundation)



대규모 협업 소프트웨어가 아닌  
단품 소프트웨어의 개발자들 권익보호

회사제품에 대한  
개발자 생태계  
단품을 위한 개발자 협업

공공의 이익을 위한 소프트웨어



Open Infrastructure  
FOUNDATION

- 2010년 NASA가 클라우드 컴퓨팅 솔루션을  
오픈하기로 결정
- 클라우드 인프라오픈소스에 열광한 대규모 벤더의 참여
- 자신의 제품을 클라우드 인프라에 맞추려는 노력들
- 핵심 프로젝트



CLOUD NATIVE  
COMPUTING FOUNDATION

- 구글이 쿠버네티스에 대한 통제권을 양도하고 이를  
리눅스 재단에서 운영하게 될 CNCF(Cloud Native  
Computing Foundation)라는 새로 설립된 재단에 기부

벤더들의 협업을 통해  
만들어진 제품

개인 프로젝트의 범위를  
넘어선 글로벌  
인프라 프로젝트

재단 차원에서 Quality를  
유지하기 위한 노력

# 모든 밴더들이 함께 지원하는 오픈소스 생태계의 진화

## OIF(Open Infrastructure Foundation (구 openstack Foundation))



인프라 수명관리



가상화 컨테이너



모든  
인프라  
관리



에지 클라우드  
인프라



게이밍 CI/CD

## CNCF(Cloud Native Computing Foundation)

### Graduated



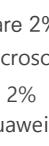
### View on the CNCF landscape →



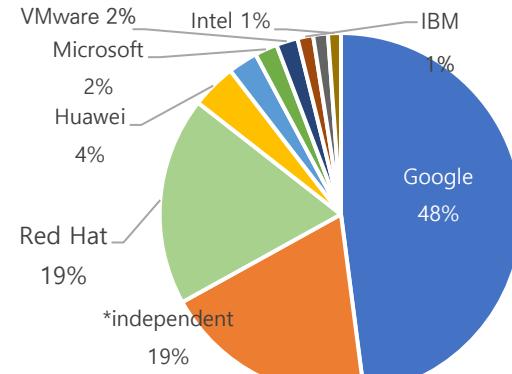
### Incubating



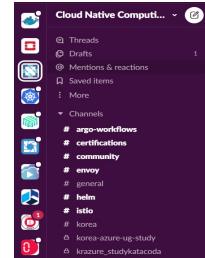
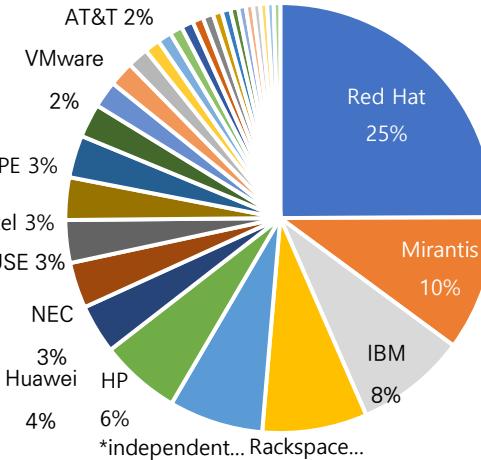
### View on the CNCF landscape →



## 구글 주도의 오픈소스 프로젝트 Kubernetes



## 인프라 벤더 주도의 오픈소스 프로젝트 오픈스택



1. 다양한 분야의 기여로 어떤 특정 밴드가 유지보수 하는 것이 불가능함

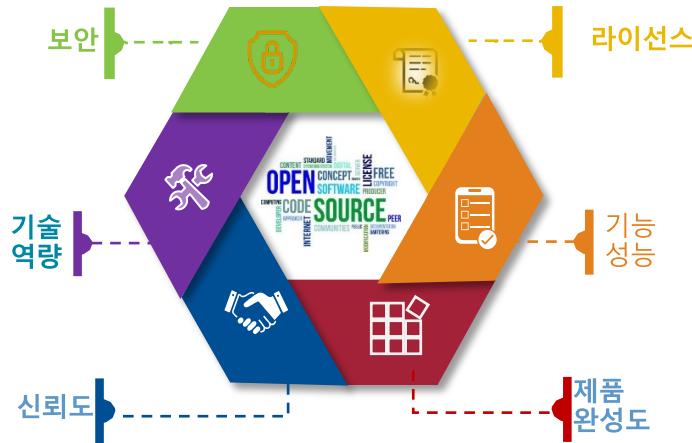
2. 특정 벤더의 제품을 추가 할 경우, 벤더 랙인이 걸림

3. 벤더 랙인이 걸릴 경우 전체 제품의 안정성과 업그레이드 담보 불가

**클라우드로 가기 위해 커뮤니티 재단의 오픈소스를 사용하기**



# OSS Infra 적용 고려사항

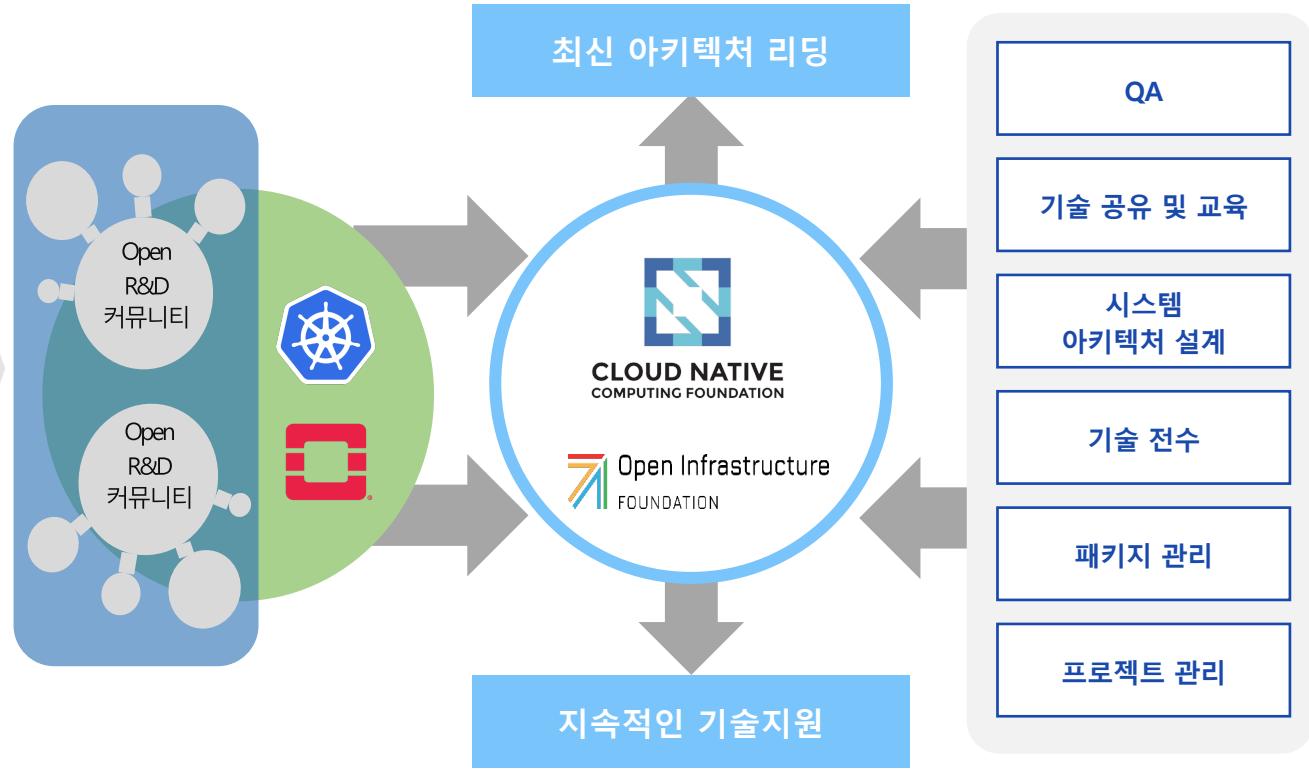


## Vendors의 오픈소스 선택시 고려사항



- 오픈소스 생태계가 만들어져야 함
  - 특정 밴드의 지원은 종속성의 문제 야기
  - 종속되는 순간 생태계는 고립됨
  - 밴더 종속성의 업그레이드 이슈
  - 안정성과 업그레이드 담보 불가

## 커뮤니티 오픈소스의 안정성과 보안취약점을 해결하기 위한 재단의 협업



139

177K+

A small icon of a document with horizontal lines, indicating a file or report.

837

48K+

시가총액 : 26,675조  
Funding : 76.2조

# 02

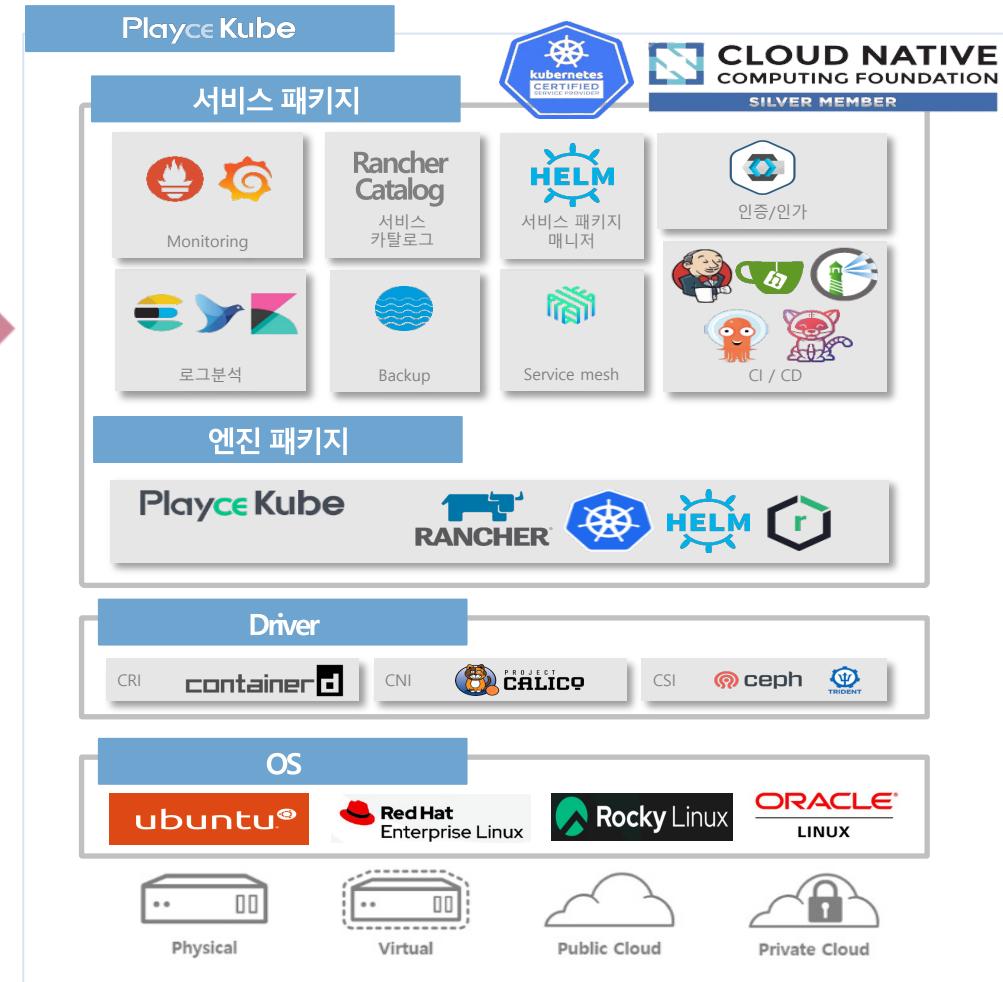
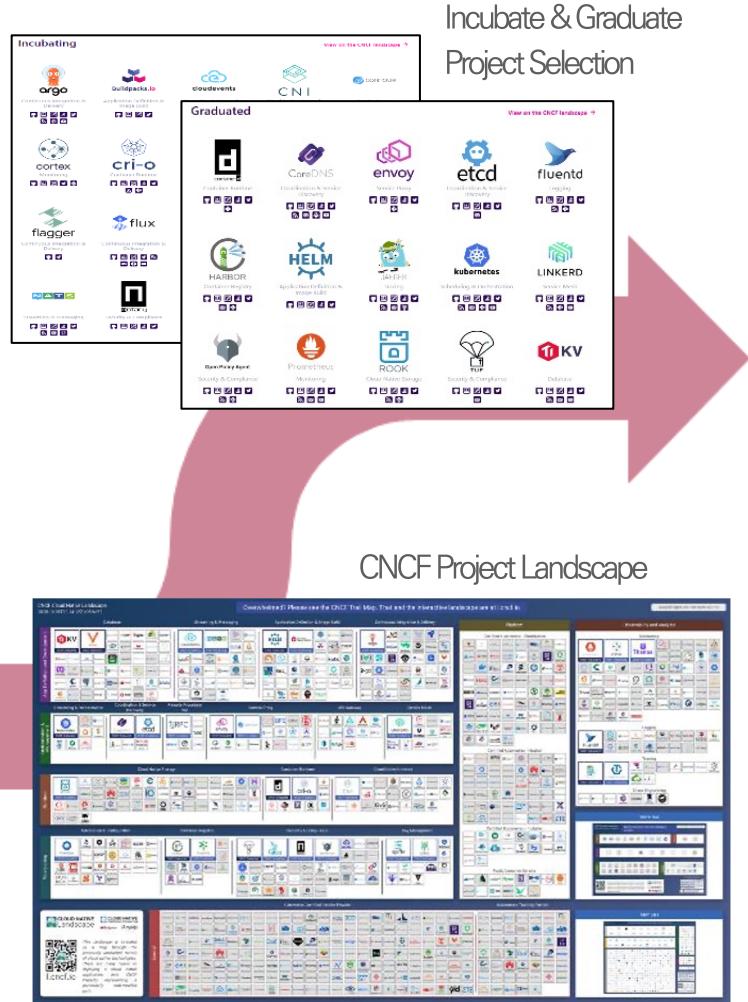
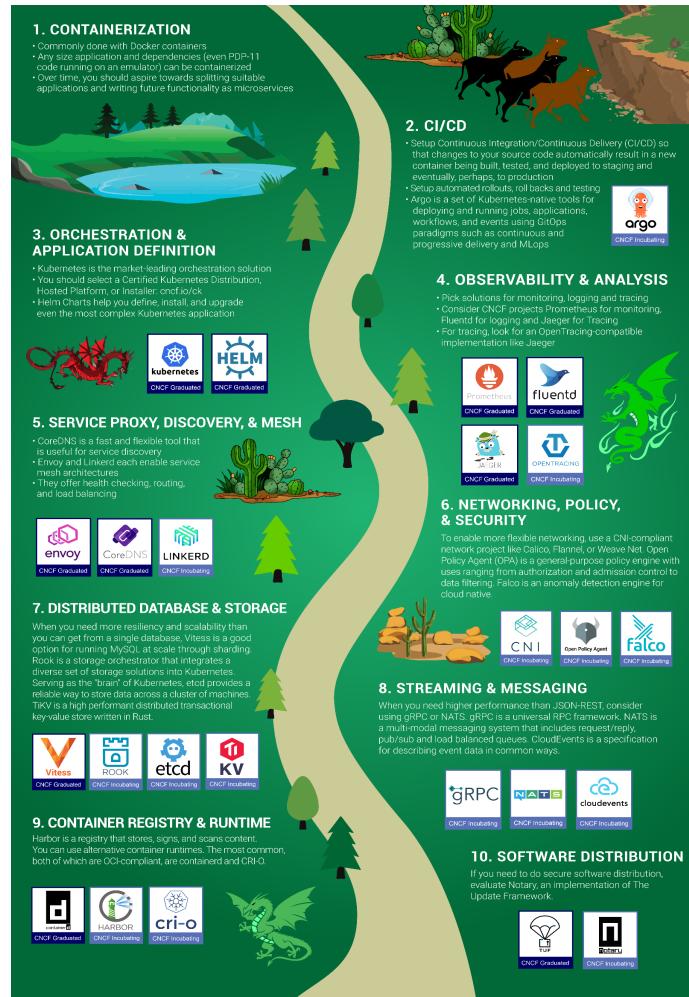
---

## 필요한 오픈소스 선택하기



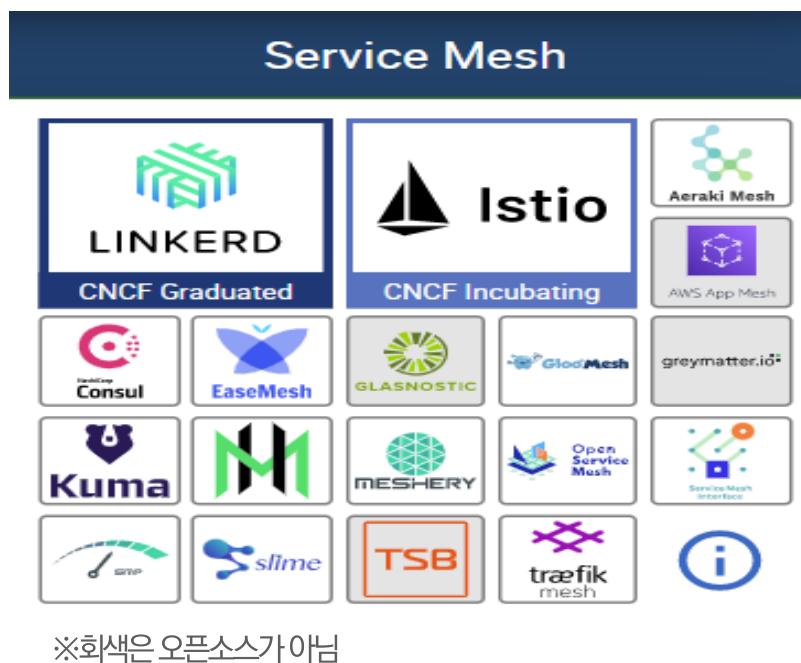
## 오픈소스 선택 과정

- 순수 오픈소스 커뮤니티 재단의 검증된 SW Stack Pool 을 선정하여 고객의 인프라에 최적화된 환경을 구성



# [예시] 오픈소스 패키지 모듈 선정 과정

- CNCF의 동일 카테고리 프로젝트 중에서 패키지에 넣을 영역의 솔루션을 비교하여, 최근에 높은 star (500개 이상) 와 활발한 활동 체크



**Linkerd**  
Cloud Native Computing Foundation (CNCF)  
Orchestration & Management · Service Mesh

Ultra light, ultra simple, ultra powerful. Linkerd adds security, observability, and reliability to Kubernetes, without the complexity.

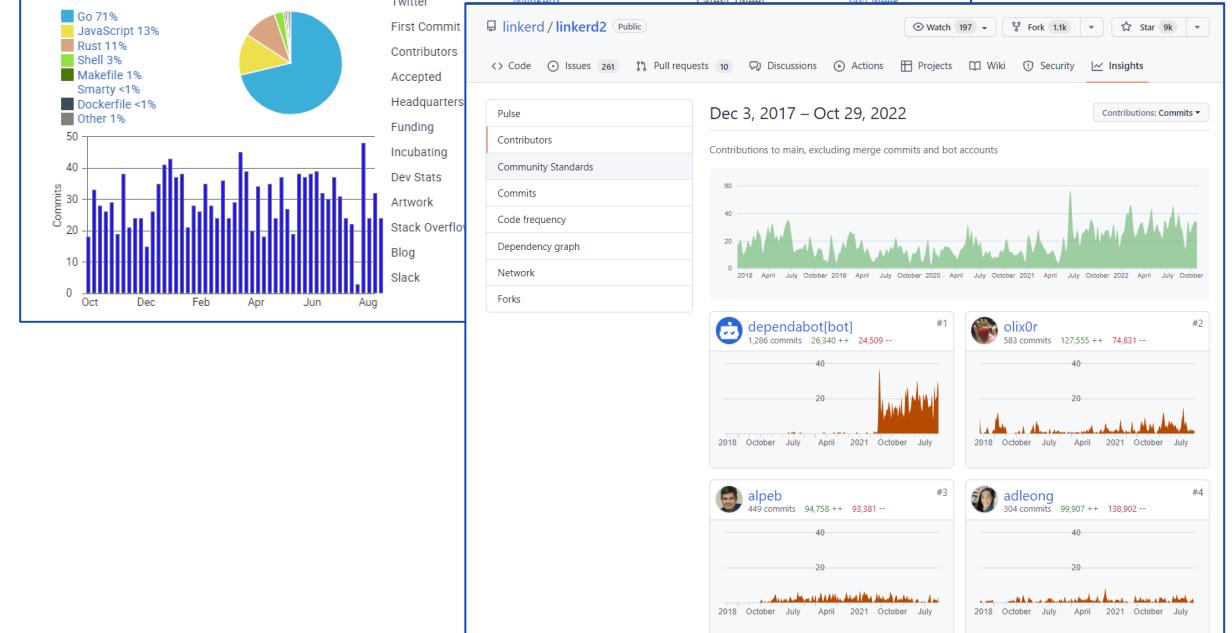
Website: [linkerd.io](https://linkerd.io)  
Repository: [github.com/linkerd/linkerd2](https://github.com/linkerd/linkerd2) ★ 8,947  
Crunchbase: [crunchbase.com/organization/cloud-native-computing-foundation](https://crunchbase.com/organization/cloud-native-computing-foundation)  
LinkedIn: [@linkerd](https://linkedin.com/company/cloud-native-computing-foundation)  
Twitter: [Latest Tweet](#) [Last week](#)

First Commit: Dec 3, 2017 · Last Commit: Oct 29, 2022

Contributors: 1926 · License: Apache License 2.0 · OpenSSF Best Practices: ✓ · Pulse: Public

Commits: Go 71%, JavaScript 13%, Rust 11%, Shell 3%, Makefile 1%, Smarty <1%, Dockerfile <1%, Other 1%

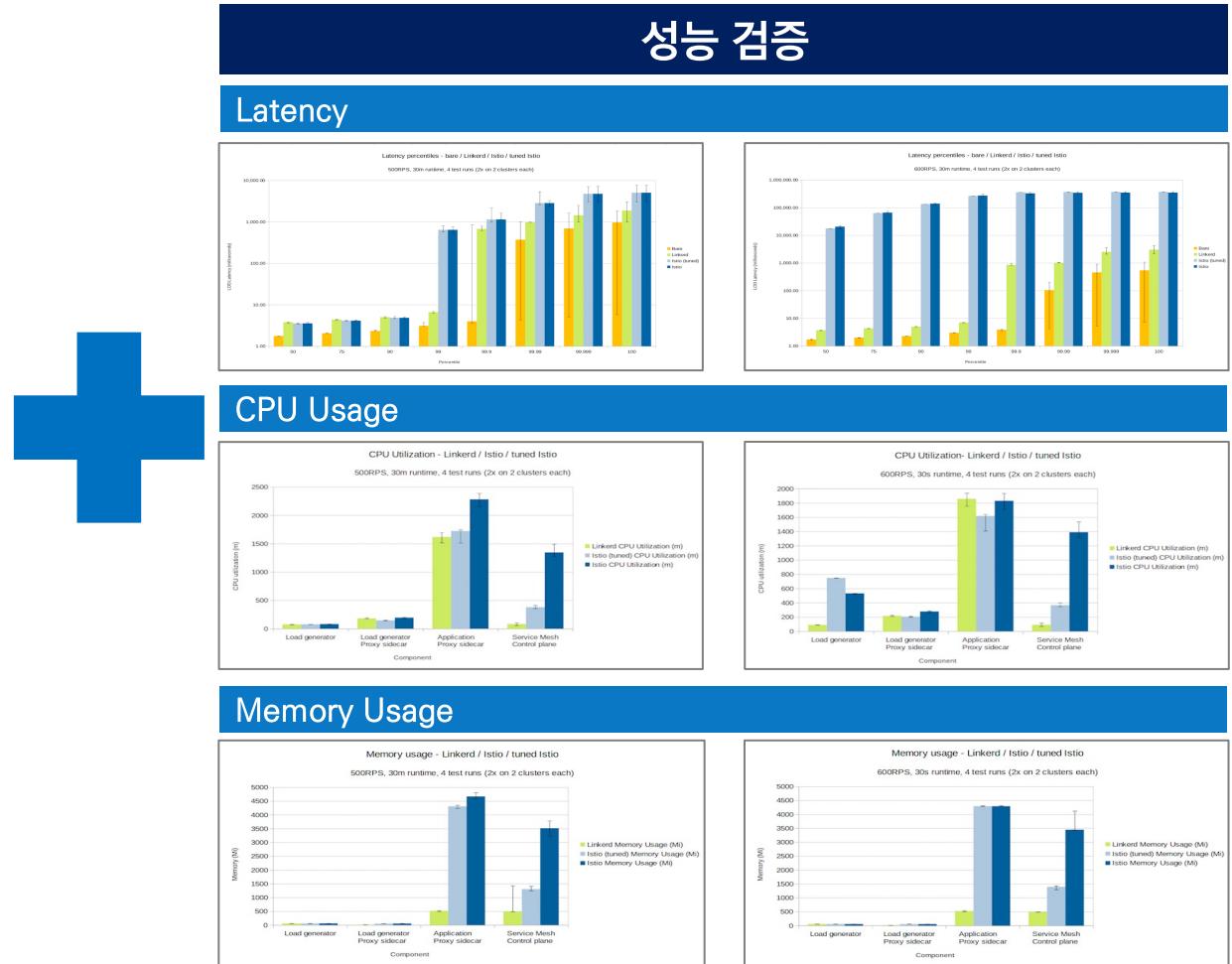
Contributions to main, excluding merge commits and bot accounts:



# [예시] 오픈소스 패키지 모듈 선정 과정

- 여러가지 자료를 확인하고, 실제 구현해서, 필요한 요소 비교작업 수행

기능 검증		
	Istio	Linkerd
지원	Kubernetes + VM	Kubernetes
Architecture	<ul style="list-style-type: none"> <li>No SPOF</li> <li>Envoy sidecar proxy</li> <li>No node-agent</li> </ul>	<ul style="list-style-type: none"> <li>No SPOF</li> <li>sidecar proxy</li> <li>No node-agent</li> </ul>
Secure Communication	<ul style="list-style-type: none"> <li>mTLS support</li> <li>Certificate Management</li> </ul>	<ul style="list-style-type: none"> <li>mTLS support</li> <li>Certificate Management</li> </ul>
Communication Protocols	<ul style="list-style-type: none"> <li>TCP / HTTP / gRPC</li> </ul>	<ul style="list-style-type: none"> <li>TCP / HTTP / gRPC</li> </ul>
Traffic Mgmt.	<ul style="list-style-type: none"> <li>Blue/Green Deployment</li> <li>Circuit Breaking</li> <li>Fault Injection</li> <li>Rate Limiting</li> </ul>	<ul style="list-style-type: none"> <li>Blue/Green Deployment</li> <li>Circuit Breaking</li> <li>Fault Injection</li> <li><b>NO Rate Limiting</b></li> </ul>
Observability	<ul style="list-style-type: none"> <li>Yes, with Prometheus</li> </ul>	<ul style="list-style-type: none"> <li>Yes, with Prometheus</li> </ul>
Multicluster	<ul style="list-style-type: none"> <li>Support</li> </ul>	<ul style="list-style-type: none"> <li>Support</li> </ul>
Operations Complexity	<ul style="list-style-type: none"> <li>High</li> </ul>	<ul style="list-style-type: none"> <li>Low</li> </ul>
결론	<ul style="list-style-type: none"> <li>가장 많은 기능과 유연성을 가지지만 복잡함</li> </ul>	<ul style="list-style-type: none"> <li><b>Kubernetes만을 위해서는 최적의 솔루션</b></li> </ul>



# [예시] 오픈소스 패키지 모듈 선정 과정

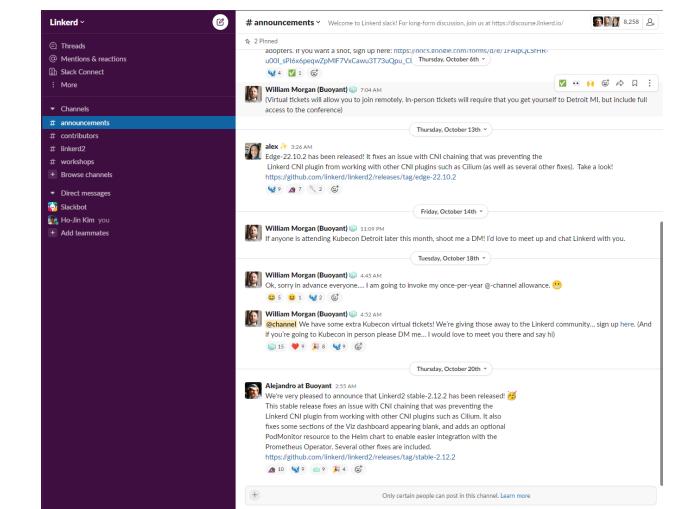
- 커뮤니티에 직접 참여하여, 현재 상황과 향후 roadmap에 대해서 고민하고, 필요시 커뮤니케이션 시도

Newsletter 구독

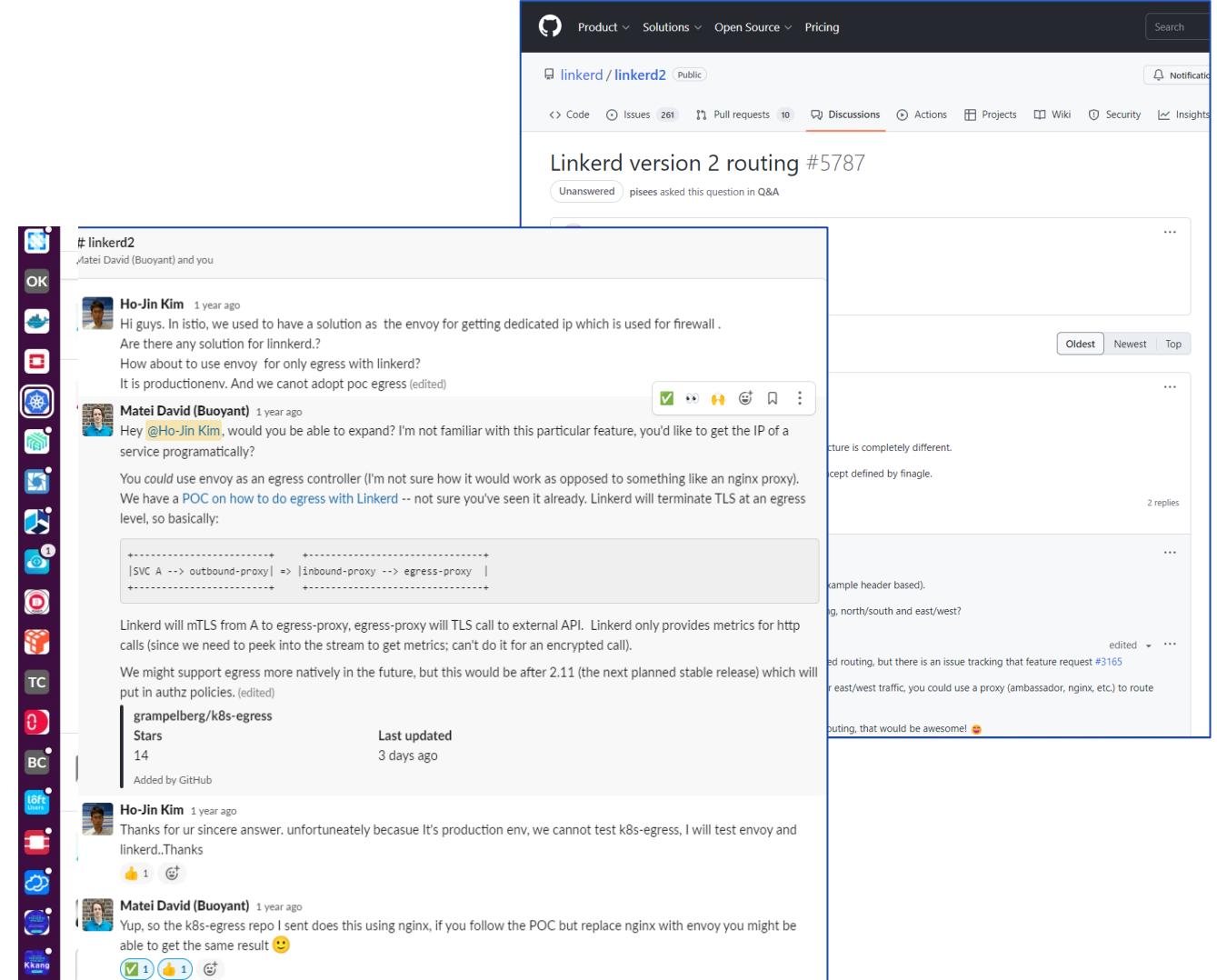


**Buoyant**  
**Service Mesh Newsletter**

A Kubernetes engineer's guide to mTLS  
Community 와 communication



The screenshot shows the #announcements channel in the Linkerd Slack workspace. It displays several messages from users like William Morgan (@buoyant) and Matei David (@buoyant) discussing the use of envoy for getting dedicated IP and the possibility of using envoy as an egress controller. The interface includes a sidebar with channels like #Threads, #Contributors, and #Linkerd2.



linkerd / linkerd2 Public

Linkerd version 2 routing #5787

Unanswered pisees asked this question in Q&A

# linkerd2

Matei David (Buoyant) and you

Ho-Jin Kim 1 year ago

Hi guys. In istio, we used to have a solution as the envoy for getting dedicated ip which is used for firewall. Are there any solution for linkerd?

How about to use envoy for only egress with linkerd?

It is productionenv. And we can't adopt poc egress (edited)

Matei David (Buoyant) 1 year ago

Hey @Ho-Jin Kim, would you be able to expand? I'm not familiar with this particular feature, you'd like to get the IP of a service programmatically?

You could use envoy as an egress controller (I'm not sure how it would work as opposed to something like an nginx proxy). We have a POC on how to do egress with Linkerd -- not sure you've seen it already. Linkerd will terminate TLS at an egress level, so basically:

```
-----+-----+
|SVC A-->outbound-proxy| -> |Inbound-proxy-->egress-proxy|
-----+-----+
```

Linkerd will mTLS from A to egress-proxy, egress-proxy will TLS call to external API. Linkerd only provides metrics for http calls (since we need to peek into the stream to get metrics; can't do it for an encrypted call).

We might support egress more natively in the future, but this would be after 2.11 (the next planned stable release) which will put in authz policies. (edited)

grampelberg/k8s-egress

Stars 14 Last updated 3 days ago

Added by GitHub

Ho-Jin Kim 1 year ago

Thanks for your sincere answer. Unfortunately because it's a production env, we cannot test k8s-egress, I will test envoy and linkerd. Thanks

Matei David (Buoyant) 1 year ago

Yup, so the k8s-egress repo I sent does this using nginx, if you follow the POC but replace nginx with envoy you might be able to get the same result 😊

1 1

This screenshot shows a GitHub discussion thread titled "Linkerd version 2 routing". The thread is public and has 261 issues and 10 pull requests. The user Ho-Jin Kim asks about using envoy for getting a dedicated IP for a firewall. Matei David responds with a proposed architecture involving envoy as an egress controller. He also mentions a POC for k8s-egress using nginx. The conversation continues with Ho-Jin Kim expressing interest in testing it and Matei David confirming it's possible using nginx as a proxy.

# 03

---

## 필요한 오픈소스 통합 패키징 방법

- Helm



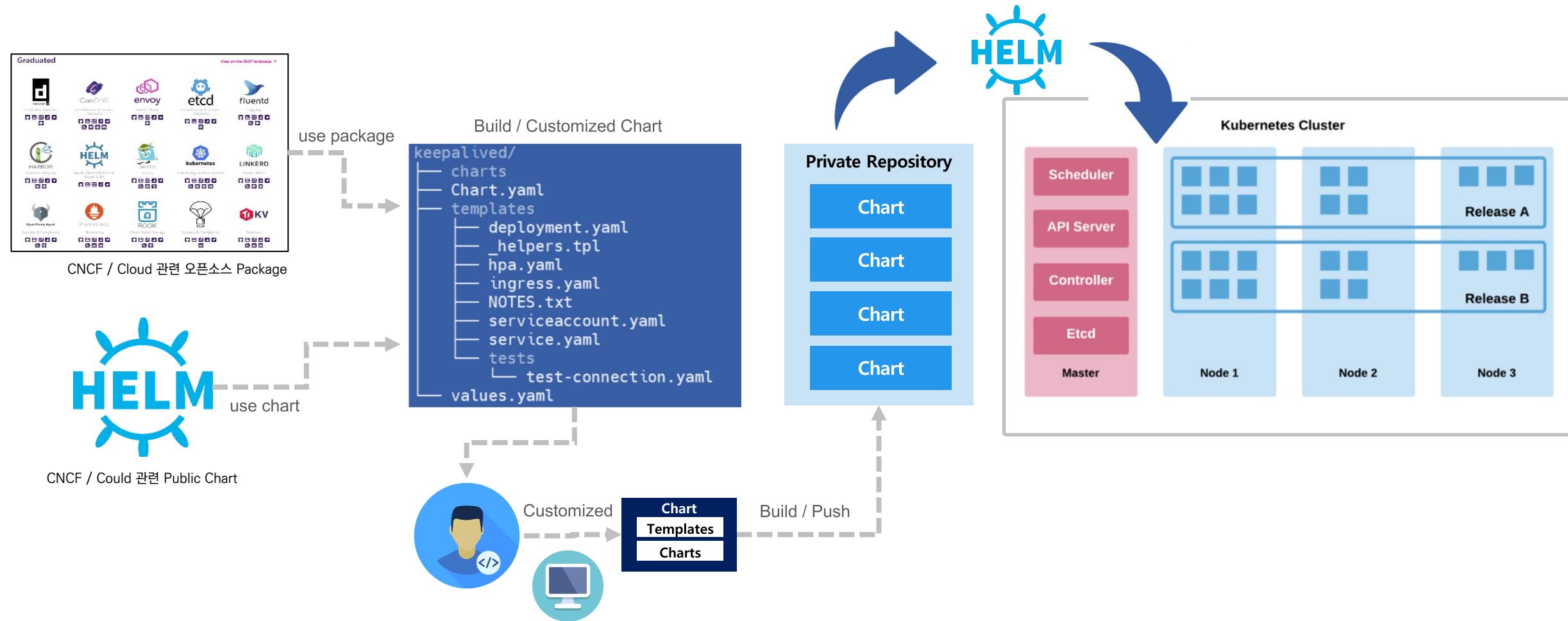
Open Infrastructure  
FOUNDATION



CLOUD NATIVE  
COMPUTING FOUNDATION

# Helm Chart로 서비스 패키징

- Cloud 관련 오픈소스를 즉시 사용 가능하도록 커스터마이즈 하여 Private Repository를 생성



# Helm Chart로 서비스 패키징

- Helm chart 생성 → 패키징 → 레포지토리 등록

1

## helm 다운로드 및 설치

```
curl -LO https://get.helm.sh/helm-v3.10.1-linux-amd64.tar.gz
tar zxf helm-v3.10.1-linux-amd64.tar.gz
mv linux-amd64/helm /usr/local/bin/helm
```

2

## 컨테이너 이미지 빌드

```
FROM docker.io/library/ubuntu:22.04

RUN apt-get update && apt-get install -y --no-install-recommends \
    keepalived && \
    apt-get clean && \
    rm -rf /var/lib/apt/lists/*

COPY run.sh /run.sh

ENTRYPOINT ["/run.sh"]
```

3

## helm 커マン드로 샘플 차트 생성

```
[root@playekube-deploy keepalived]# helm create keepalived
Creating keepalived
[root@playekube-deploy keepalived]# tree keepalived/
keepalived/
├── charts
│   └── Chart.yaml
└── templates
    ├── deployment.yaml
    ├── _helpers.tpl
    ├── hpa.yaml
    ├── ingress.yaml
    ├── NOTES.txt
    ├── serviceaccount.yaml
    ├── service.yaml
    └── tests
        └── test-connection.yaml

3 directories, 10 files
```

4

## Helm chart 수정

```
# helm chart info change
sed -i "s/description: .*/description: A Helm chart for keepalived/" keepalived/Chart.yaml
sed -i "s/appVersion: .*/appVersion: \"2.2.4\"/" keepalived/Chart.yaml
sed -i "1i\annotations:" keepalived/Chart.yaml
sed -i "/^annotations:/a\ catalog.cattle.io/certified: keepalived/Chart.yaml
sed -i "1i\keywords:" keepalived/Chart.yaml
sed -i "/^keywords:/a\-\ infrastructure" keepalived/Chart.yaml
```

5

## Yaml file 작성

```
cat << EOF > keepalived/templates/daemonset.yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: {{ include "keepalived.fullname" . }}
  labels:
    {{- include "keepalived.labels" . | nindent 4 --}}
spec:
  selector:
    matchLabels:
      {{- include "keepalived.selectorLabels" . | nindent 6 }}
```

6

## values.yaml 파일 수정

```
cat << EOF > keepalived/values.yaml
image:
  repository:
    registry.local.cloud:5000/ubuntu/keepalived
  pullPolicy: IfNotPresent
  tag: v2.2.4

  keepalived:
    vrouter_id: "150"
    vrouter_interface: "eth0"
    virtual_ip: "192.168.0.250"
```

# Helm Chart로 서비스 패키징

- Helm chart 생성 → 패키징 → 레포지토리 등록



```
[root@playcekube-deploy keepalived]# helm search repo playcekube
NAME          CHART VERSION   APP VERSION
playcekube/argo-cd      4.9.11        v2.4.3
playcekube/argo-events    2.0.3         v1.7.1
playcekube/argo-rollouts  2.18.0        v1.2.0
playcekube/csi-driver-nfs v4.0.0         v4.0.0
playcekube/curl           0.1.0         7.80.0
playcekube/gitea          5.0.9         1.16.8
playcekube/gitlab         6.1.1         15.1.1
playcekube/harbor          14.0.3        2.5.2
playcekube/ingress-nginx   4.1.4         1.2.1
playcekube/jenkins         4.1.12        2.346.1
playcekube/keepalived     0.1.0         2.2.4
playcekube/keycloak        7.1.17        16.1.1
playcekube/knative          v1.6.0        v1.6.0
playcekube/kube-prometheus-stack 35.3.1        0.56.2
playcekube/kubeapps         9.0.3         2.4.6
playcekube/kubernetes-dashboard 5.4.1         2.5.1
playcekube/linkerd-jaeger   2.11.1        stable-2.11.1
playcekube/linkerd-multicloud 2.11.1        stable-2.11.1
playcekube/linkerd-viz      2.11.1        stable-2.11.1
playcekube/linkerd2         2.11.1        stable-2.11.1
playcekube/metrics-server    3.8.2          0.6.1
playcekube/minio             4.0.2          RELEASE.2022-05-08T23-50-31Z
playcekube/oauth2-proxy      6.2.0          7.2.0
playcekube/opendistro-es     1.13.3        1.13.3
playcekube/tekton-pipeline   v0.32.1        v0.32.1
playcekube/velero            2.30.1        1.9.0
[root@playcekube-deploy keepalived]#
[root@playcekube-deploy keepalived]#
```

DESCRIPTION
A Helm chart for Argo CD, a declarative, GitOps...
A Helm chart for Argo Events, the event-driven ...
A Helm chart for Argo Rollouts
CSI NFS Driver for Kubernetes
A Helm chart for curlimages
Gitea Helm chart for Kubernetes
The One DevOps Platform
Harbor is an open source trusted cloud-native r...
Ingress controller for Kubernetes using NGINX a...
Jenkins - Build great things at any scale! The ...
A Helm chart for keepalived
Keycloak is a high performance Java-based ident...
A Helm chart for knative
kube-prometheus-stack collects Kubernetes manif...
Kubeapps is a web-based UI for launching and ma...
General-purpose web UI for Kubernetes clusters
The Linkerd-Jaeger extension adds distributed t...
The Linkerd-Multicloud extension contains res...
The Linkerd-Viz extension contains observabilit...
Linkerd gives you observability, reliability, a...
Metrics Server is a scalable, efficient source ...
Multi-Cloud Object Storage
A reverse proxy that provides authentication wi...
Open Distro for Elasticsearch
A Helm chart for tekton-pipeline
A Helm chart for velero

## 서비스 패키지 구성

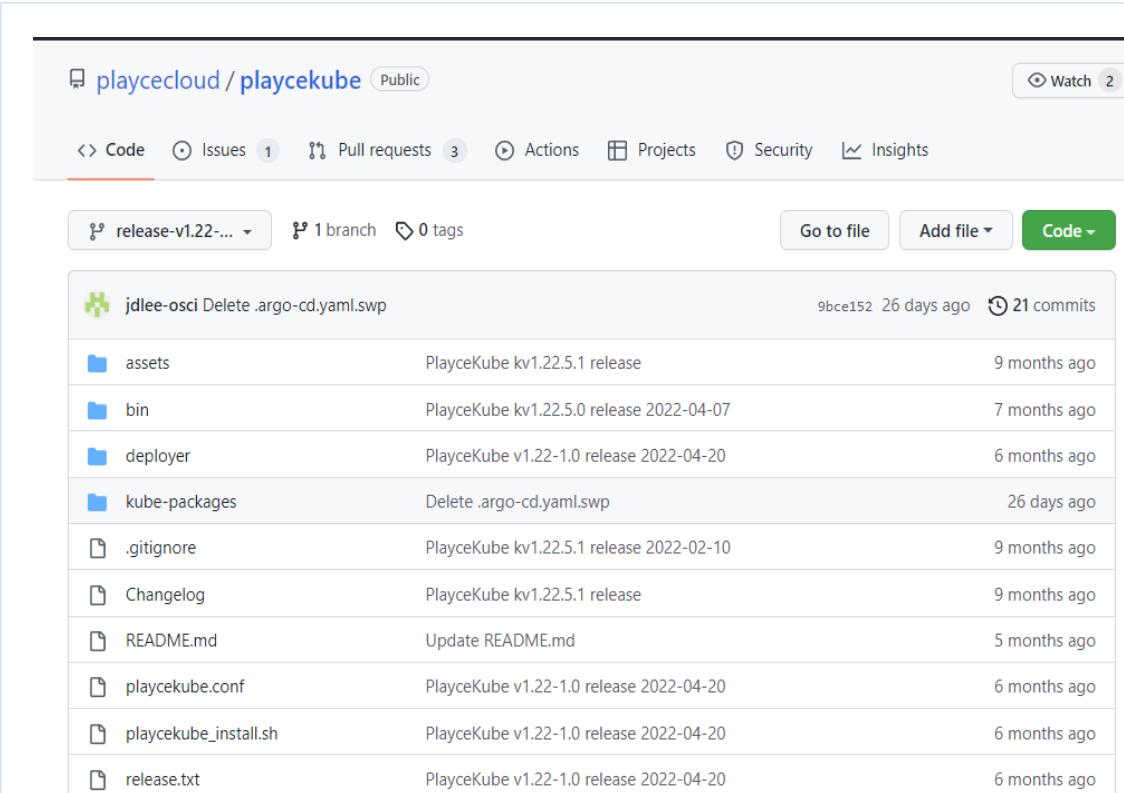
## 솔루션 주요기능

- 운영 환경에 맞는 패키지 카탈로그를 기반으로 사용목적에 맞는 클러스터 환경 구성



# Github – public github : playcekube

- <https://github.com/playcecloud/playcekube>



playcecloud / playcekube Public

Code Issues 1 Pull requests 3 Actions Projects Security Insights

release-v1.22-... 1 branch 0 tags Go to file Add file Code

jdlee-osci Delete .argo-cd.yaml.swp 9bce152 26 days ago 21 commits

assets	PlayceKube kv1.22.5.1 release	9 months ago
bin	PlayceKube kv1.22.5.0 release 2022-04-07	7 months ago
deployer	PlayceKube v1.22-1.0 release 2022-04-20	6 months ago
kube-packages	Delete .argo-cd.yaml.swp	26 days ago
.gitignore	PlayceKube kv1.22.5.1 release 2022-02-10	9 months ago
Changelog	PlayceKube kv1.22.5.1 release	9 months ago
README.md	Update README.md	5 months ago
playekube.conf	PlayceKube v1.22-1.0 release 2022-04-20	6 months ago
playekube_install.sh	PlayceKube v1.22-1.0 release 2022-04-20	6 months ago
release.txt	PlayceKube v1.22-1.0 release 2022-04-20	6 months ago



README.md

이 매뉴얼은 베어메탈 환경에서 Kubernetes를 기본으로 한 Playce Kube을 설치할 수 있는 사용자 가이드입니다.  
PlayceKube는 테스트 환경에서 컨테이너를 배포하기 위해 구축된 오픈소스 컨테이너 관리 플랫폼이며, 본 솔루션을 사용하면 다양한 컴퓨팅 환경에서 Kubernetes를 보다 쉽게 설치하고 실행 할 수 있습니다.

Playce Kube를 설치하는 기본 환경은 Deploy Node 1대, Master Node 1대, Worker Node 1대를 기준으로 설치를 진행합니다.

## Playce Kube 설치

### 최소/권장 요구 조건(Single Host 사양)

항목	권장사양	최소사양
CPU	8 Core	8 Core
Memory	16 GB	4 GB
HDD	200 GB >	200 GB
OS	Rocky Linux 8.5	Rocky Linux 8.5

위 사양 수준의 3대가 기본적으로 구성되어야 하며, Rocky Linux 8.5를 해당 서버의 OS로 설치, 구성되도록 합니다.

3. 클러스터를 설치합니다.

-e 옵션으로도 개별 환경 설정을 넣을 수 있습니다

```
$ cd /playcecloud/PlayceKube/  
$ /playcecloud/PlayceKube/deployer/scripts/playcekube_kubespray.sh -f kubespray.env
```



# 04

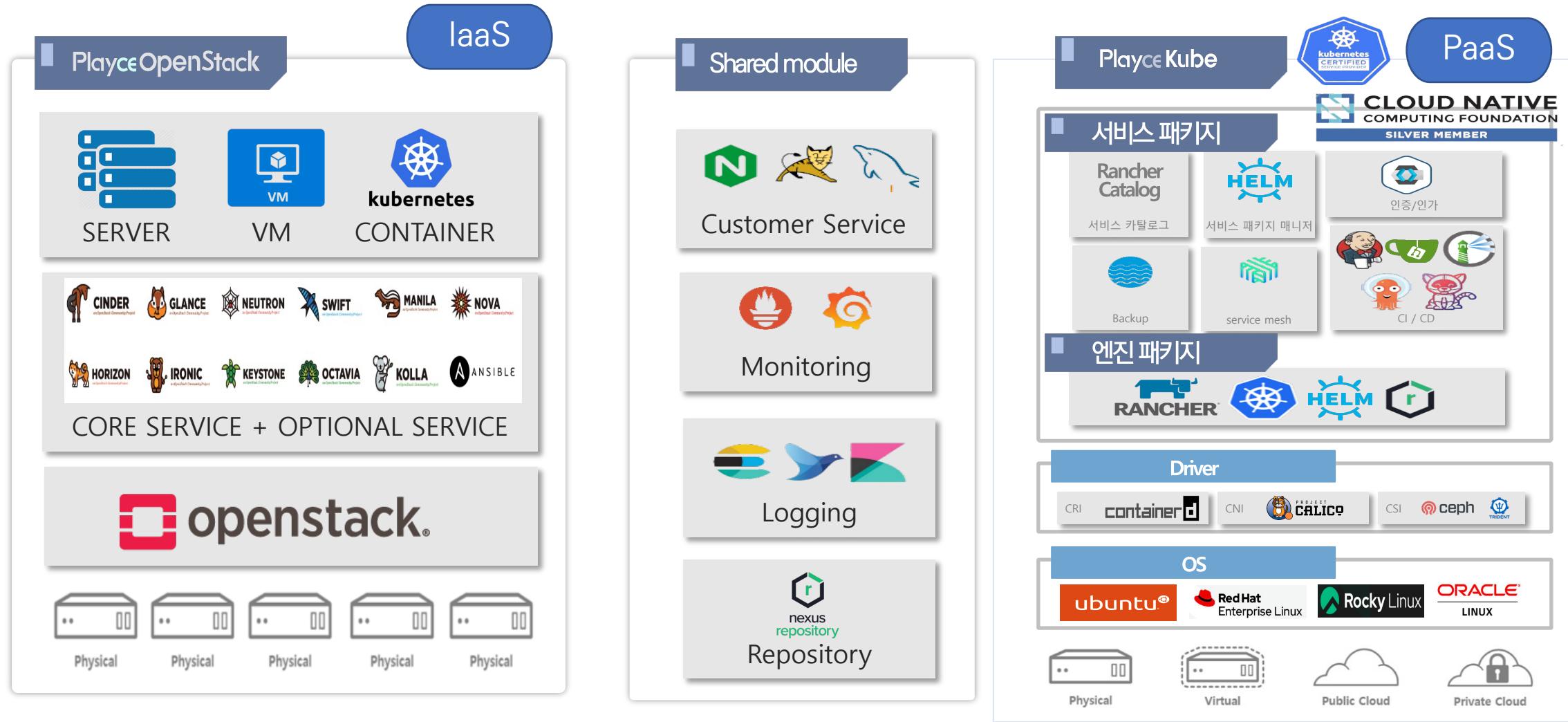
---

## 오픈소스컨설팅 패키징 개요



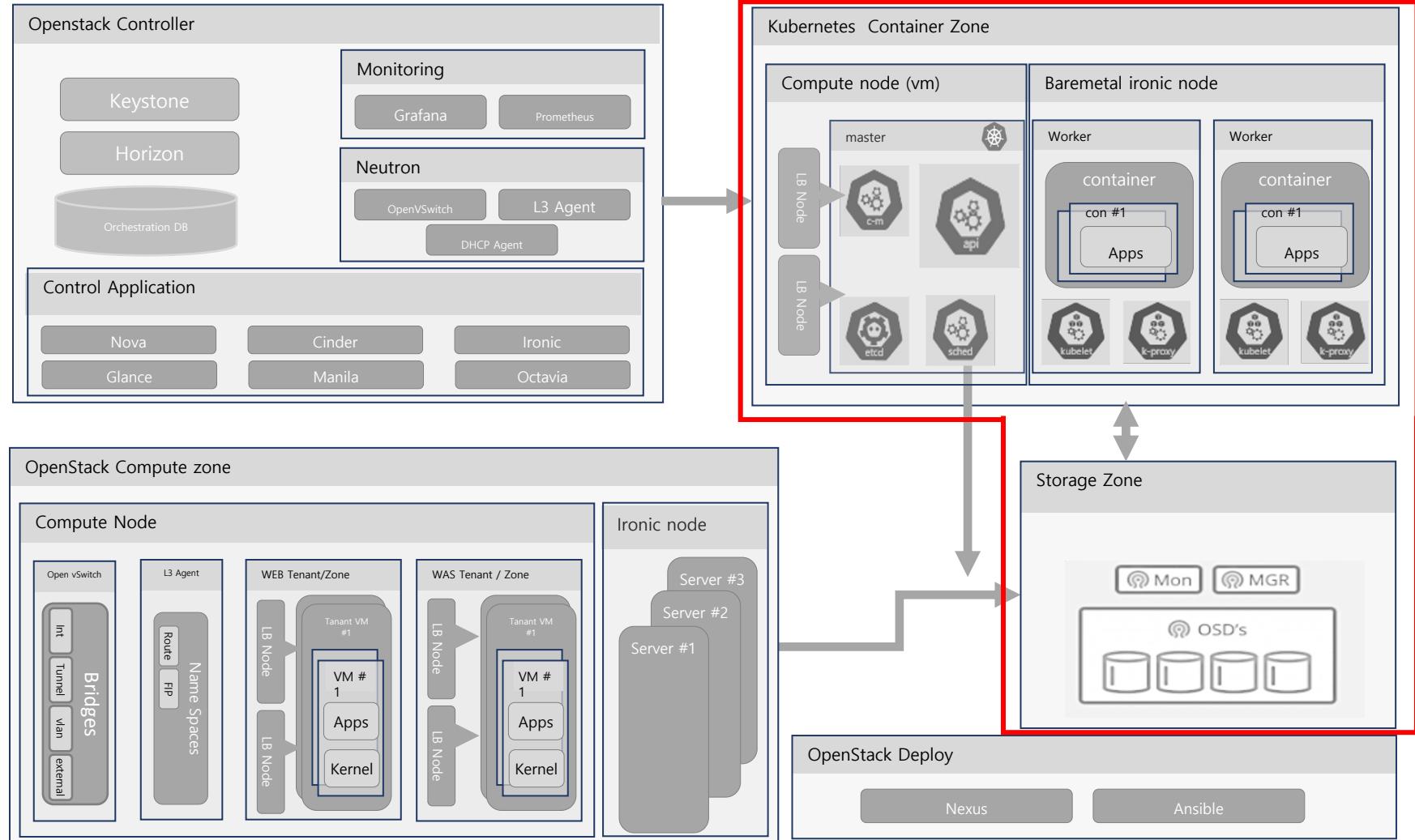
# 오픈소스컨설팅 오픈소스 패키징 - 순수 커뮤니티 오픈소스로만 구성

- 순수 오픈소스 커뮤니티 재단의 검증된 SW Stack Pool 을 선정하여 고객의 인프라에 최적화된 환경을 구성



# IaaS와 연계한 Kubernetes 구성 표준 모델

- VM과 컨테이너와 베어메탈등의 인프라를 모으고, 그것을 관리하며, 같은 스토리지 / 같은 네트워크를 사용하게 만들어 주는 SDDC 인프라



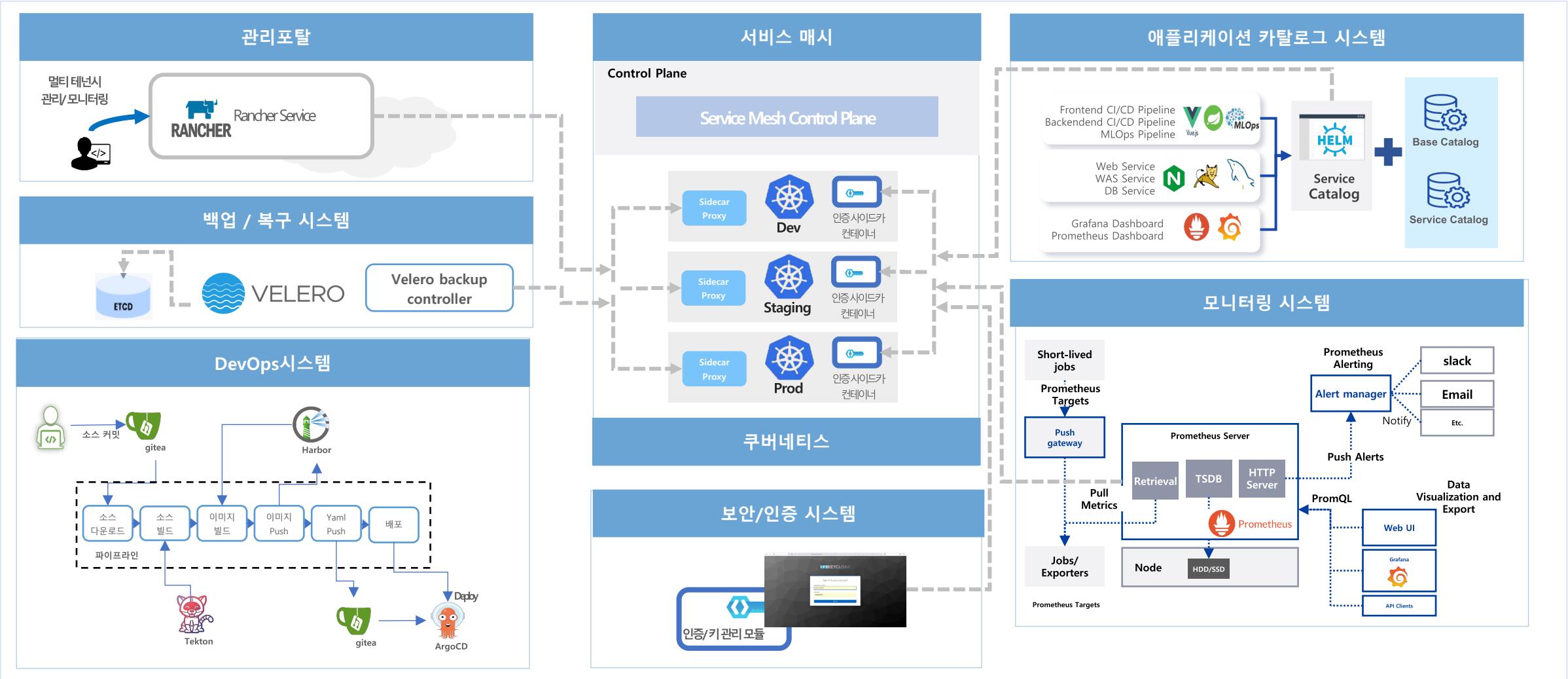
## 인프라 구축 내용 (선택 가능)

- 전체 인프라를 컨트롤할 IaaS solution (SDC\* -베어메탈/VM/컨테이너 SDN\* -L4 router / L4,7 LB / security SDS\* -RBD / NFS(cephfs)/ Object Storage)
- 필요시 언제든 확장가능한 PaaS Solution 높은 성능을 위해 baremetal에 worker구성가능
- 순수 커뮤니티 기반 소스로 업그레이드시 커뮤니티 소스 100% 반영 가능
- 오픈스택의 경우, on-line upgrade와 장애시 조치 편이성을 위해 모두 컨테이너로 패키징 필요
- Database나 보안툴을 위한 관리가능한 baremetal을 관리가능
- 오픈소스 기반의 모니터링/ADMIN 툴 제공

\* SDC (software Defined Compute)/ SDN(software Defined Network)/ SDS(Software Defined Storage) / RBD(Raw Block Device)

# 오픈소스컨설팅 PaaS 패키지 구성

- 오픈소스를 이용하여, 개발 및 시스템 Integration 지원을 위한 환경을 구현하고 있습니다.



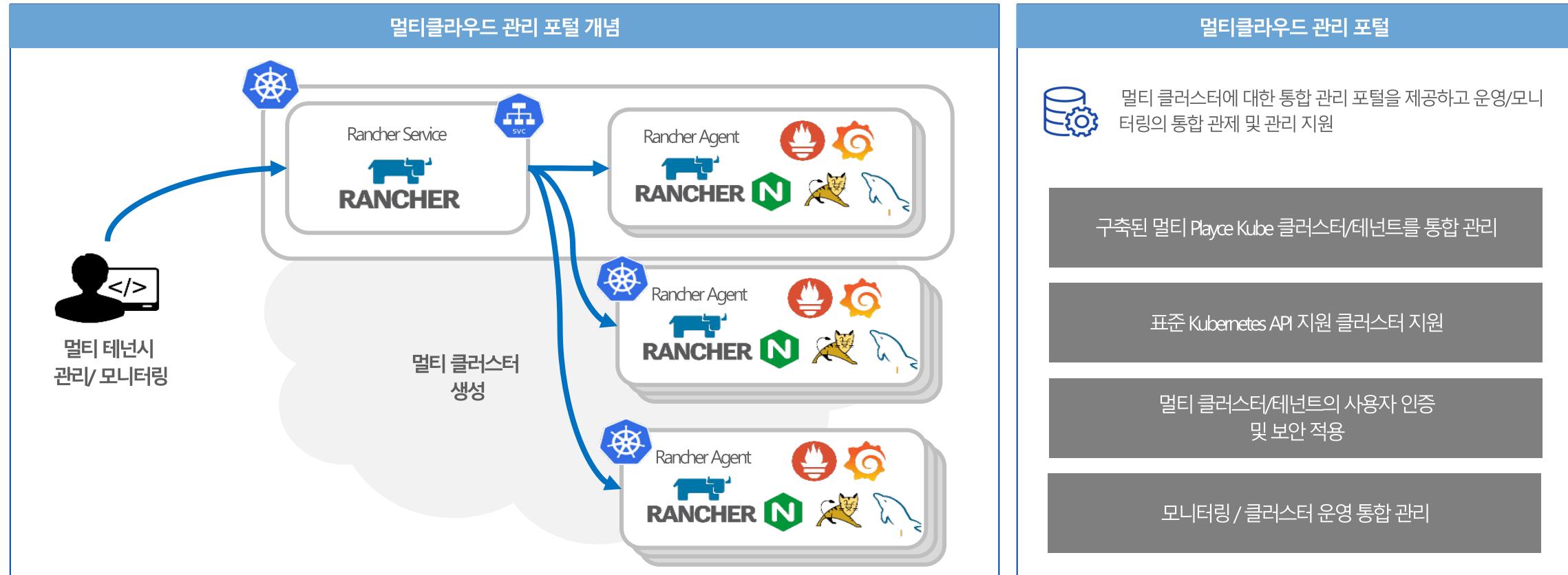
# [참고] 오픈소스컨설팅 PaaS 패키지 구성

기능	설명	선정 이유
대시보드/포탈	Rancher	멀티클러스터 관리를 지원하며, 포탈 화면 제공
모니터링	Prometheus / Grafana	많은 사용자 보유하여 많은 서비스에 모니터링 메트릭을 지원하며, 다양한 대시보드 예제가 있으며 사용자가 원하는 포맷으로 변경 가능
Storage	NFS or Vendor driver	NFS를 기본적으로 지원하며 범용적이어서 기본 장착이 용이
소스 버전 관리	Bitbucket	아틀라시안 제품 많은 사용자 보유
	Gittea	가볍고 필요한 git 서버 기능은 모두 지원
빌드 라이브러리 관리	Nexus	성능 및 가용성이 높으며 많은 사용자 보유
웹 서버	Nginx	대용량 트래픽 처리에 용이
Code Inspection	Sonarqube	다양한 프로그래밍 언어의 정적 코드 검사 및 DevOps 연동 지원
도커 레지스트리	harbor	제품은 조금 무겁지만 플러그인으로 UI 및 helm chart 지원 이미지 보안 검사 기능을 지원
배포 툴	ArgoCD	클라우드 베이스의 배포 지원 및 다양한 배포방식(Rollout, Blue/Green, Canary) 및 플러그인 지원
빌드 파이프라인	Tekton	Kubernetes 자원을 기반으로 구축되어 자원 확장 및 구성/관리가 용이
	Jenkins	범용적으로 많이 사용하는 제품으로 많은 플러그인과 파이프라인 예제가 있음
백업	Velero	object storage 기반 백업. 별다른 대안이 없음
Service Mesh	Linkerd	istio와 같은 기능은 대부분 지원하면서 사이드카로 동작하는 proxy가 가벼워 시스템 부하가 적음
Logging	ELK or EFK	범용적으로 많이 사용되고 dashboard template이 많음
오케스트레이션	Kubernetes	버전별 K8S에서 기본 제공
패키지 매니저	Helm	Kubernetes 패키지 관리
서비스 카탈로그	Rancher apps	패키지 명세서를 이용해서 손쉽게 애플리케이션을 배포하고 관리

# 쿠버네티스(Kubernetes) 기반 컨테이너 플랫폼



클라우드내 배포서버를 통해서 멀티 클러스터를 배포 함, 또한 멀티 테넌트와 클러스터를 위한 통합 관리 포털 제공 합니다

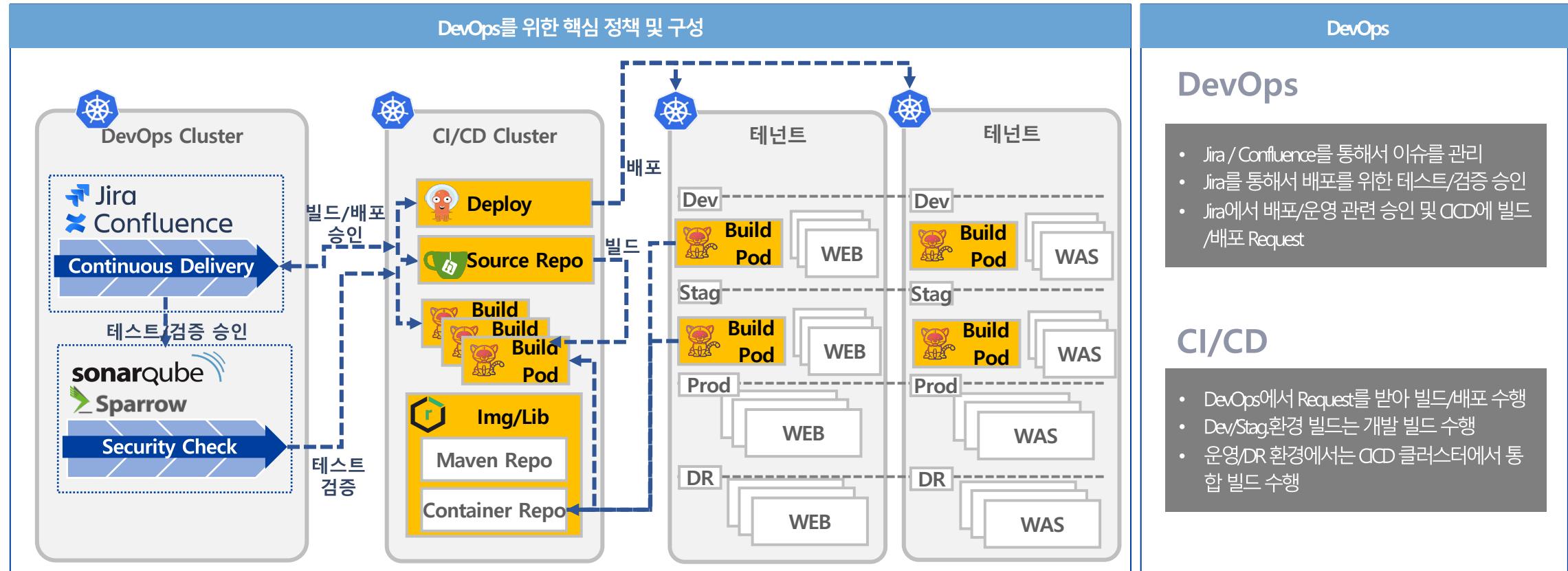


## 설계 포인트

- IaaS와 연동을 통한 테넌트 Provisioning 및 포털을 통한 통합 제어

# DevOps CI/CD 환경 구성

클러스터 기반의 별도의 CICD 클러스터를 구성하여 Continuous Integration 을 수행하고 운영/검증/개발 클러스터로 Continuous Deploy 수행

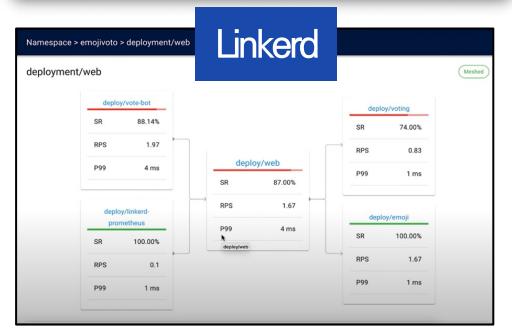
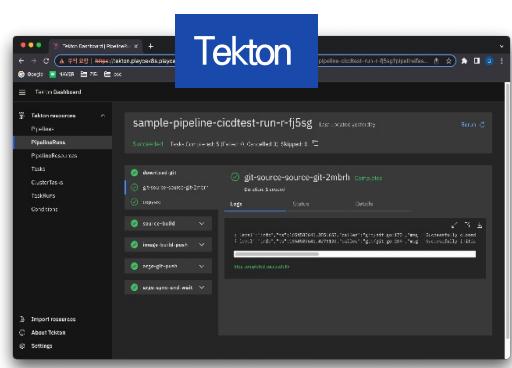
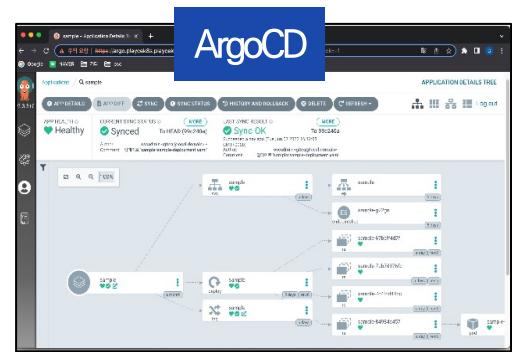
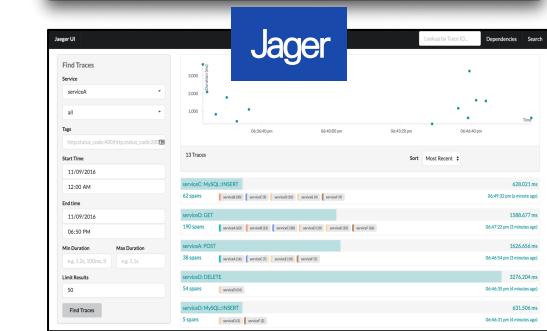
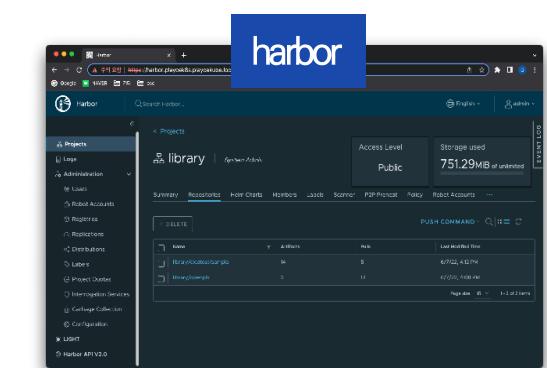
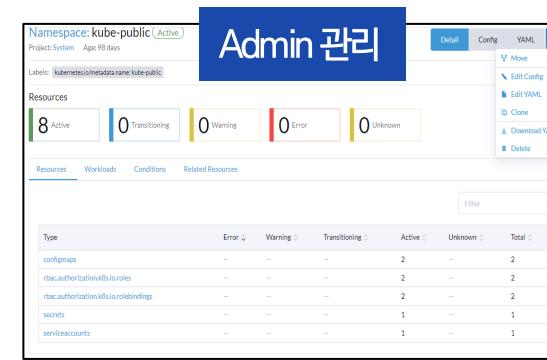
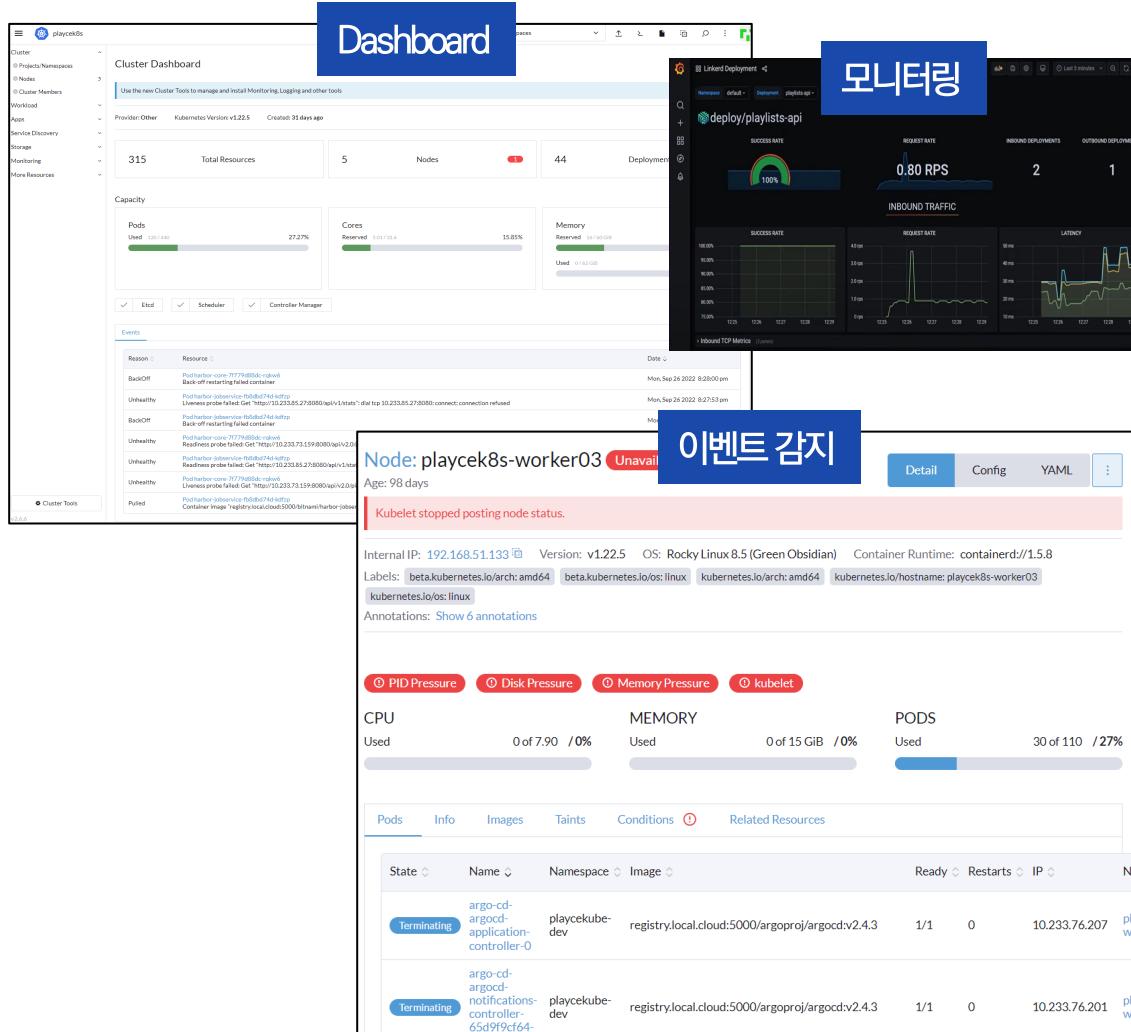


## 설계 포인트

- 고객에게 필요한 승인단계를 넣고, 개발 /스테이징에서는 각자 pod에서 개발 빌드를 수행함.
- 운영/DR 환경에서는 CICD 클러스터에서 통합 빌드 수행

# 화면 구성

각기 화면을 Keycloak SSO 연동하였으며, 필요시 승인/기타 프로세스를 위해서 jira/confluence와 연동하여 구성합니다.



# 05

---

## 실제 인프라 구현 사례



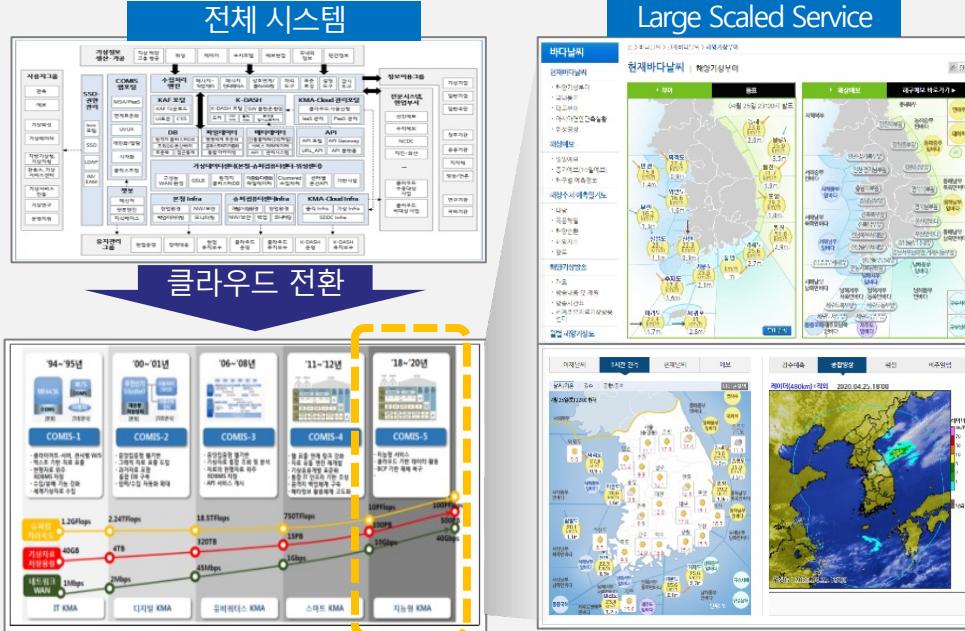
# 구축 고객 사례

구분	업체	내용	Playce Openstack	Playce Kube	수행기간	규모
금융	XX	Kubernetes 구축 컨설팅 및 DevOps 환경 구축 (vmware기반)		○	2021	100대
금융	XX은행	Kubernetes와 DevOps 환경 구축		○	2021~2022	60대
공/금융	XX	XX 클라우드 시스템 구축	○	○	2021	100대
IT	XX	소프트웨어센터 하이브리드 클라우드		○	2021	60대
IT	XX	사내 인프라시스템을 VMware와 openstack 2가지 표준으로 구성	○		2021	20대
공공	XX	XX 종합기상정보시스템(COMIS-5)의 OpenStack/Kubernetes 기반 클라우드 구축	○	○	2021-2022	250대
금융	XX	하이브리드(AWS- OpenStack)으로 가상화폐거래소 구축	○	○	2019	60대
IT	XX	미란티스 OpenStack에서 OpenStack으로 마이그레이션 및 컨설팅	○		2019	100대
공공	XX	XX 종합기상정보시스템(COMIS-5)에 기 구축된 밴더 PaaS를 마이그레이션		○	2019	60대
IDC	XX	GPU 가상화 서버 IaaS 서비스 구축	○		2019	50대
금융	XX	가상화폐거래소 구축	○		2018	23대

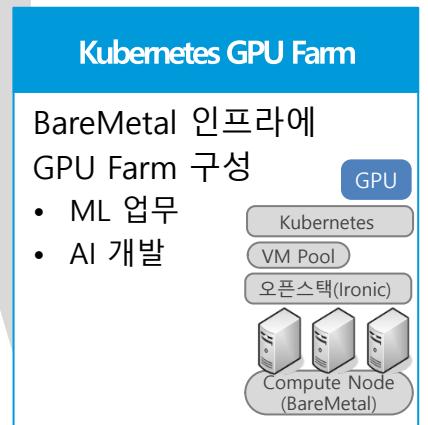
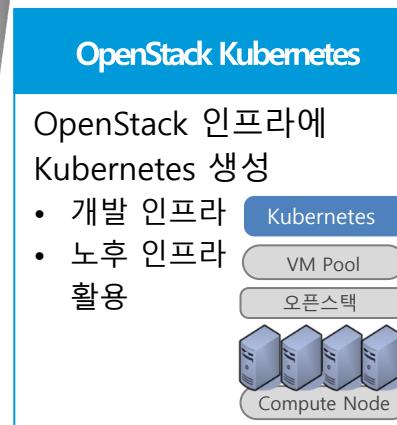
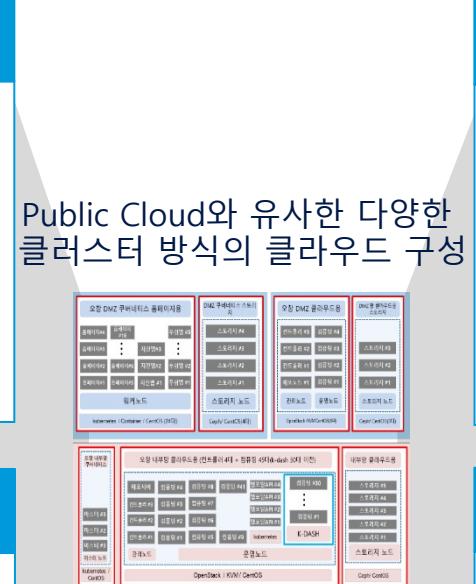
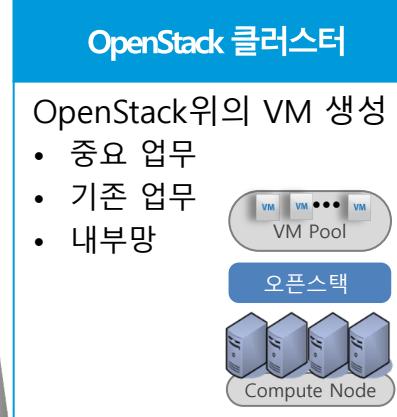
# Public형 모델의 IaaS / PaaS Private cloud 구축 사례

- Public형 모델의 IaaS / PaaS Private cloud 구축 사례로 다양한 용도의 클러스터를 구성하고 해당 클러스터를 업무 및 환경에 맞게 Zone을 나눠서 운영

## ▶ 프라이빗 오픈스택 / 쿠버네티스 사례



- 확장 가능한 오픈스택 / 쿠버네티스로 변경
- 클라우드 기반 전산자원 확충 및 운영 환경 조성
- 전산자원 할당·회수 자동화 및 시스템 부하에 따른 자동 시스템 확장 용이
- 오픈소스(오픈스택, 도커)로 대체하여 기상정보시스템 환경으로 최적화
- 타사의 IaaS / PaaS → Playce Cloud로 대체

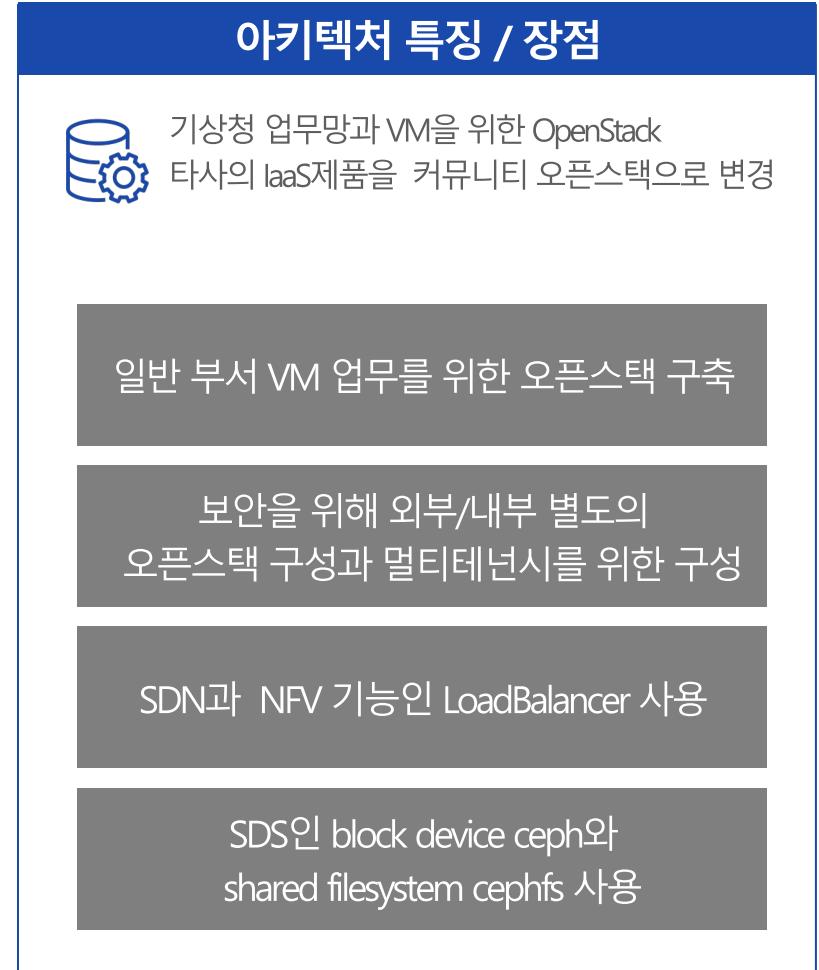
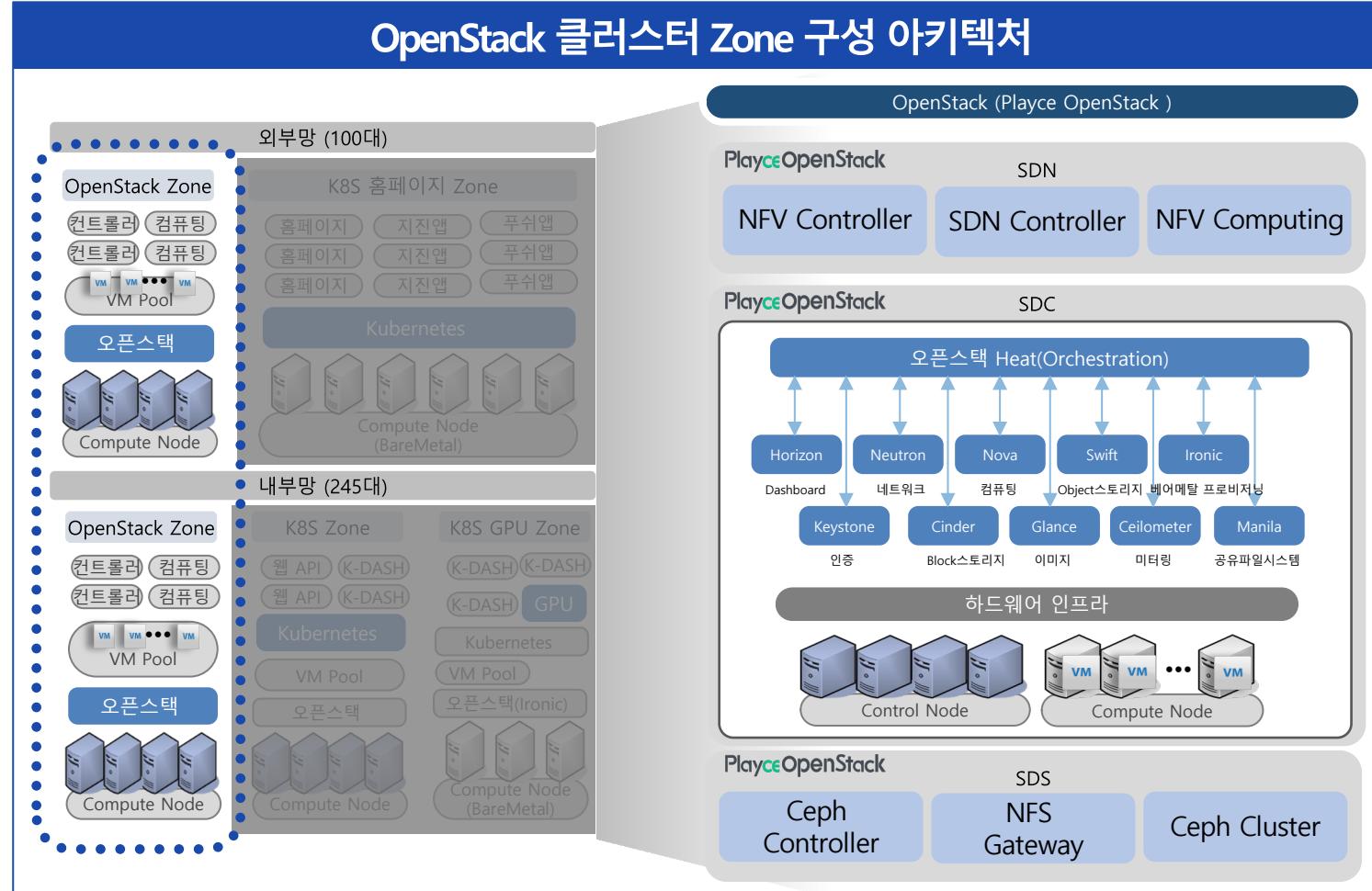


프로젝트 기간 : 2019.10 ~ 2019.12 / 2020.8~2020.12 / 2021~

\*다양한 용도의 클러스터를 Public Cloud 모델이라 칭함

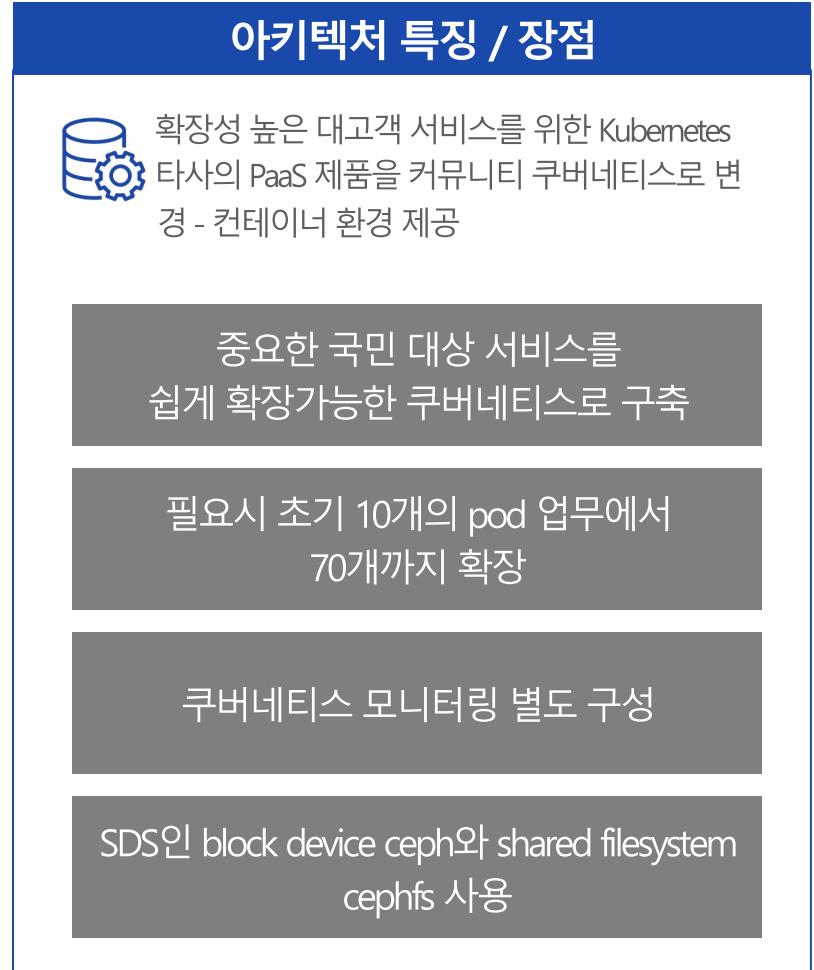
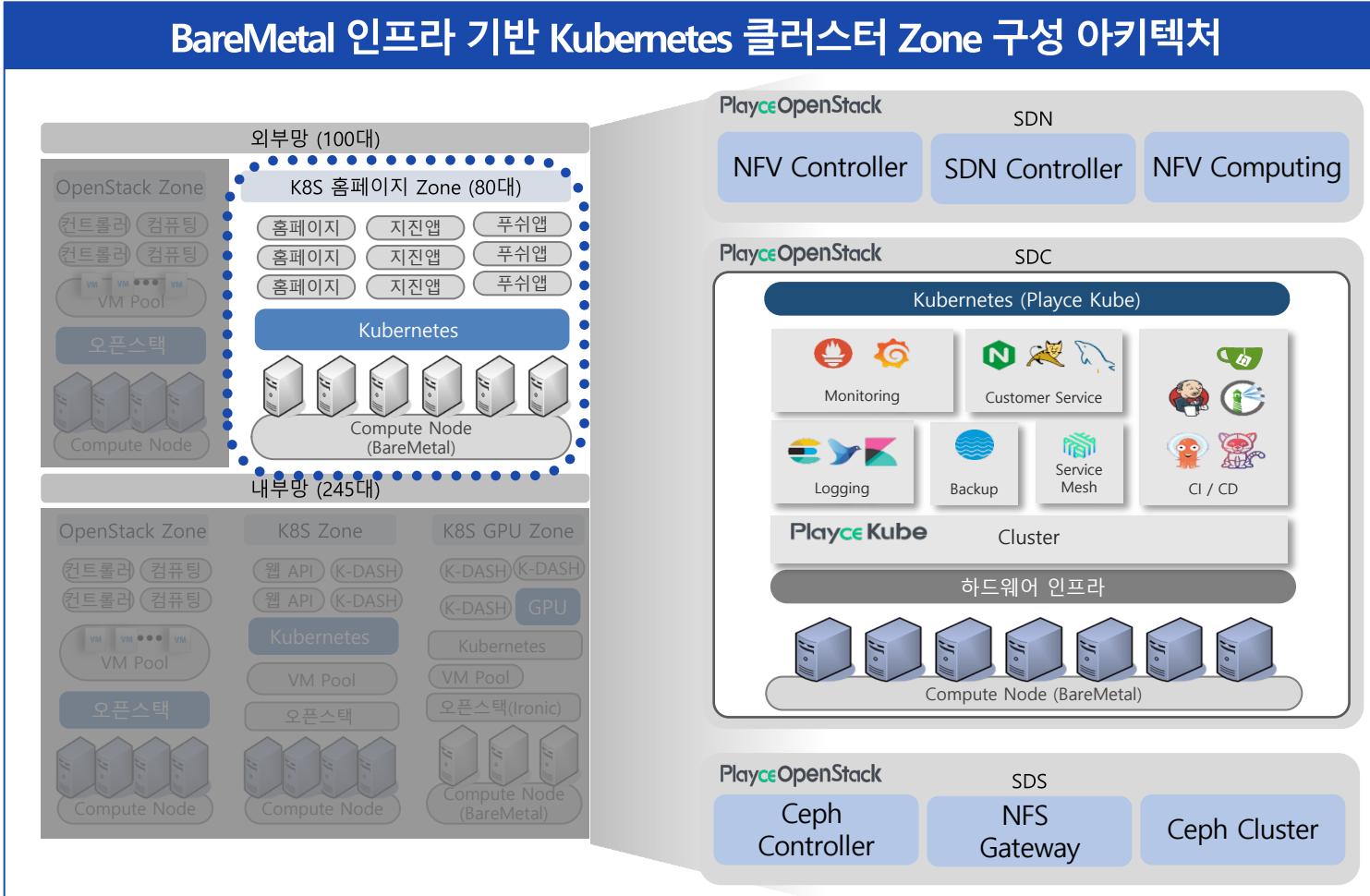
# 구축 상세 사례 - OpenStack 클러스터

- 내/외부 업무에 대해서 OpenStack Zone을 구성하여 VM 기반의 인프라 구성



# 구축 상세 사례 - BareMetal 인프라 기반의 Kubernetes

- 고객사의 Large Scale Service에 대해서 Kubernetes 환경을 구성

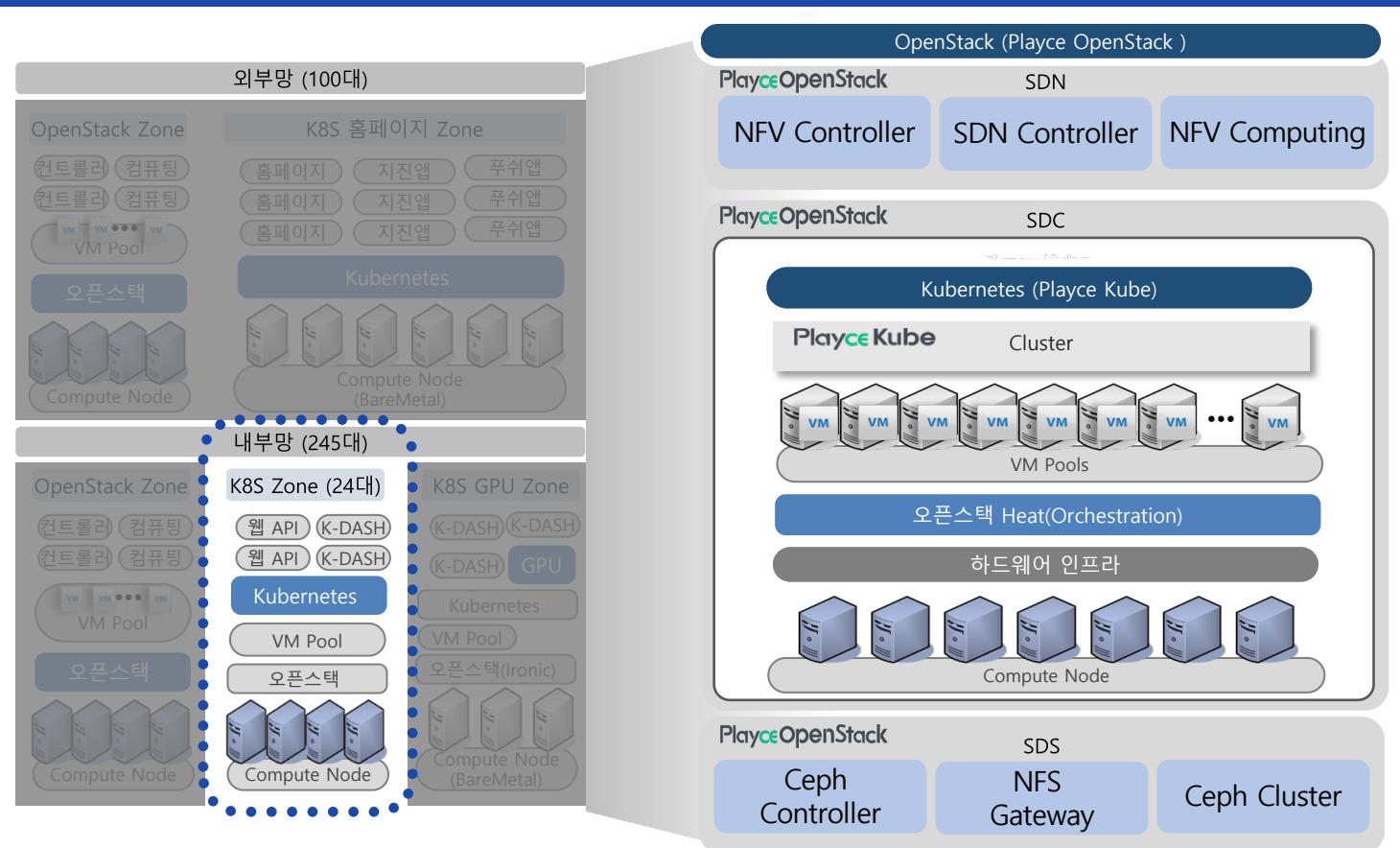


# 구축 상세 사례 - OpenStack 인프라 기반의 Kubernetes

- 고객사의 개발 환경 및 테스트 환경에 대해서 OpenStack 인프라 환경에 K8S 구축하여 K8S 노드를 확장 및 노후 장비 재활용으로 구성



## OpenStack 클러스터 기반 Kubernetes Zone 구성 아키텍처

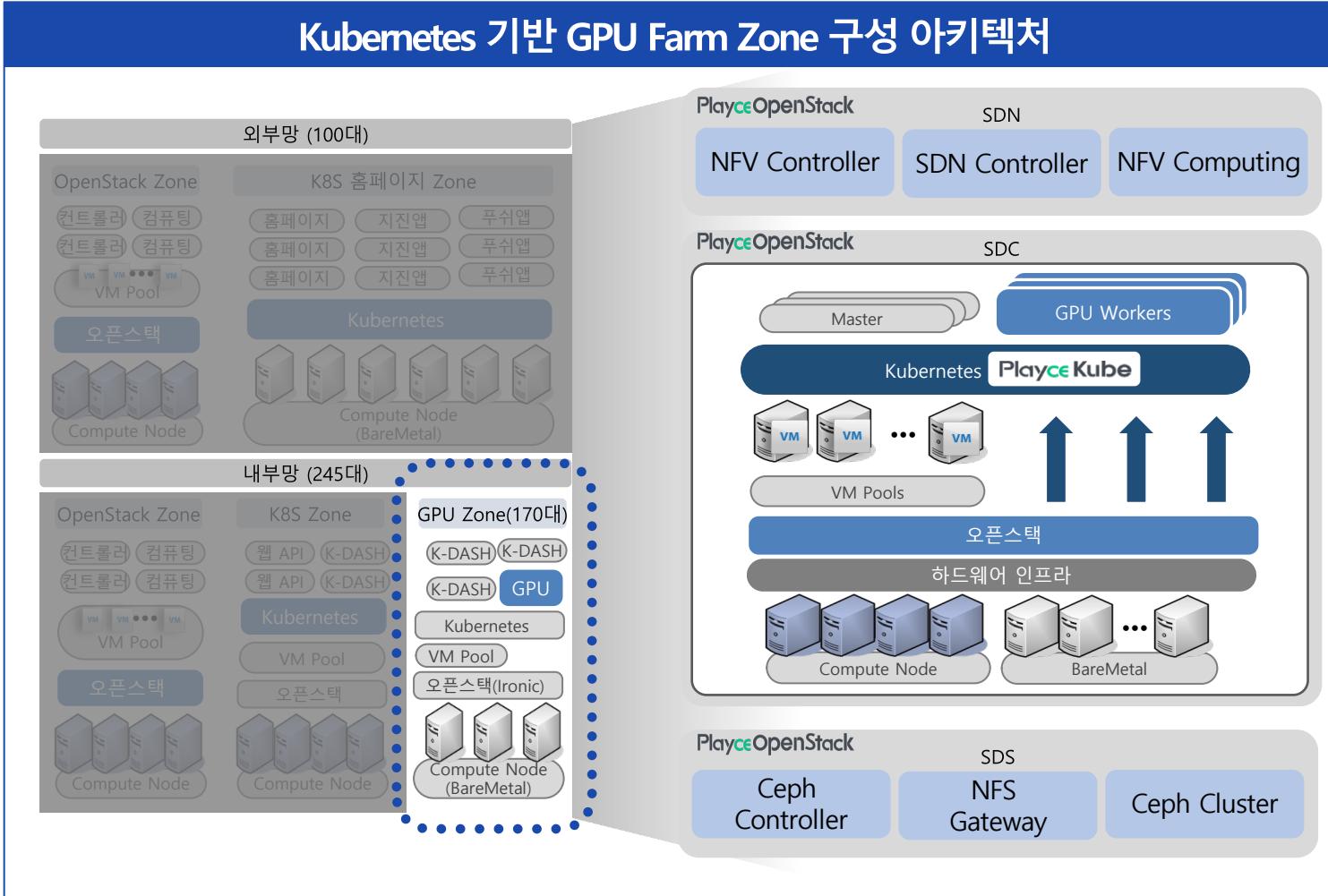


## 아키텍처 특징 / 장점

-  기상청 업무망 vm과 연계를 위한 Kubernetes
- VM위에 쿠버네티스 구성
- 멀티 클러스터 쿠버네티스 구성
- 필요시 VM단에서 쉽게 확장 가능
- 테넌트 내 VM과 연계하여 서비스 제공

# 구축 상세 사례 -Kubernetes 기반 GPU Farm / Big Data Platform

- 고객사의 AI/ML 개발 및 테스트를 위해 Kubernetes 기반의 GPU Farm/ Big Data Platform을 구성하여 리소스의 가용성 및 확장성을 지원

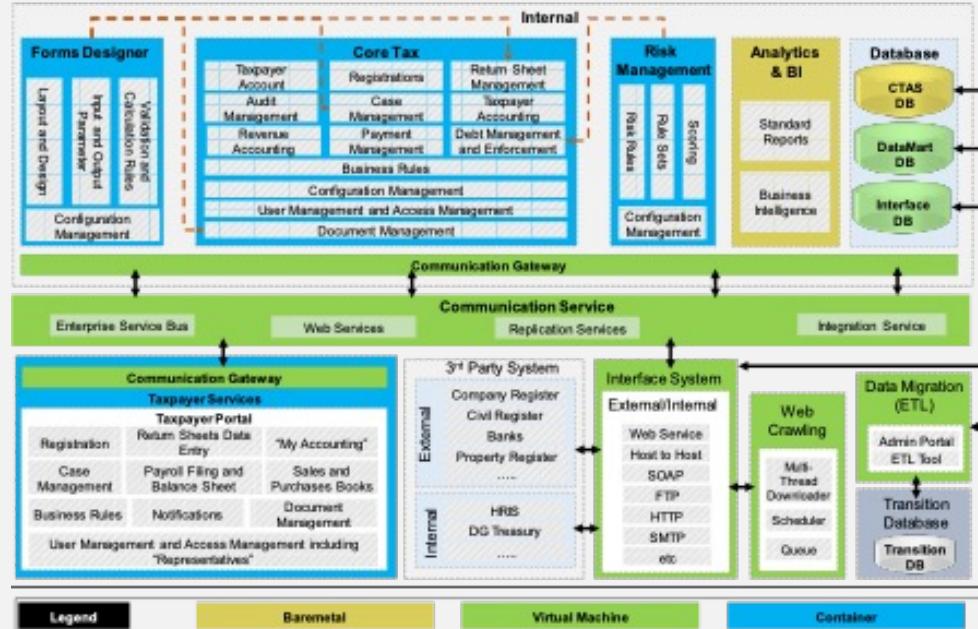


# 대규모 운영/개발/DR 클러스터 구축 사례 - 오픈소스컨설팅 해외 구축사례



대규모 개발/DC/DR 환경에 대한 특성을 구분하여 Zone을 구성하고 효율적인 백업 아키텍처 구성

## 해외 클라우드 아키텍처 구성



■ 운영: 업무별로 VM과 Container로 구분 하여 최소 이중화 구축

■ 개발: 오픈스택 기반의 VM 기반의 K8s구성, ironic으로 DB 구성

■ QA: 운영환경내에 서버와 스토리지를 병합하여 구축

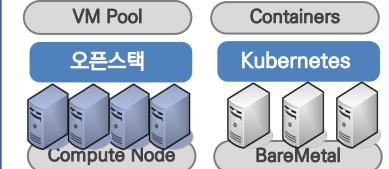
■ OpenStack과 K8s 대응 가능한 백업 구성

■ Cloud 기반의 DevOps환경 구축

## 다용도 클러스터 구성

### OpenStack / K8s Zone 구성

- OpenStack: 3rd 파티, DB
- K8s: 업무 서비스

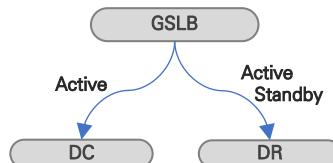


## 대규모 서비스 처리를 위한 클러스터 Zone 구성

## DR 클러스터 구성

### 장애 대비를 위한 DR구성

- 지역 특성을 고려한 DR 구성 (지형, 지진, 화산)



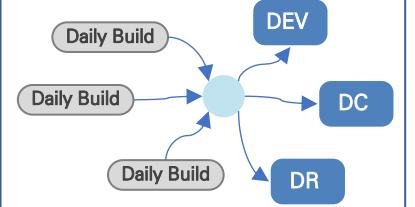
“다양한 사용자 용도와 환경으로 구성된 운영 클러스터 구성”

프로젝트 기간 : 2021.1~2021.4 / 2021.8~현재  
IaaS/PaaS 구축 운영환경/DR 컨설팅 및 구축 진행중

## CI/CD 구성

### K8s환경을 활용한 지속 배포

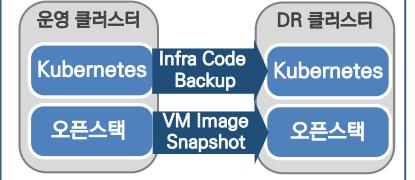
- 일일 빌드 및 즉시 배포 환경 제공



## 원거리 클러스터 백업 구성

### 서비스 별 원격 백업 정책

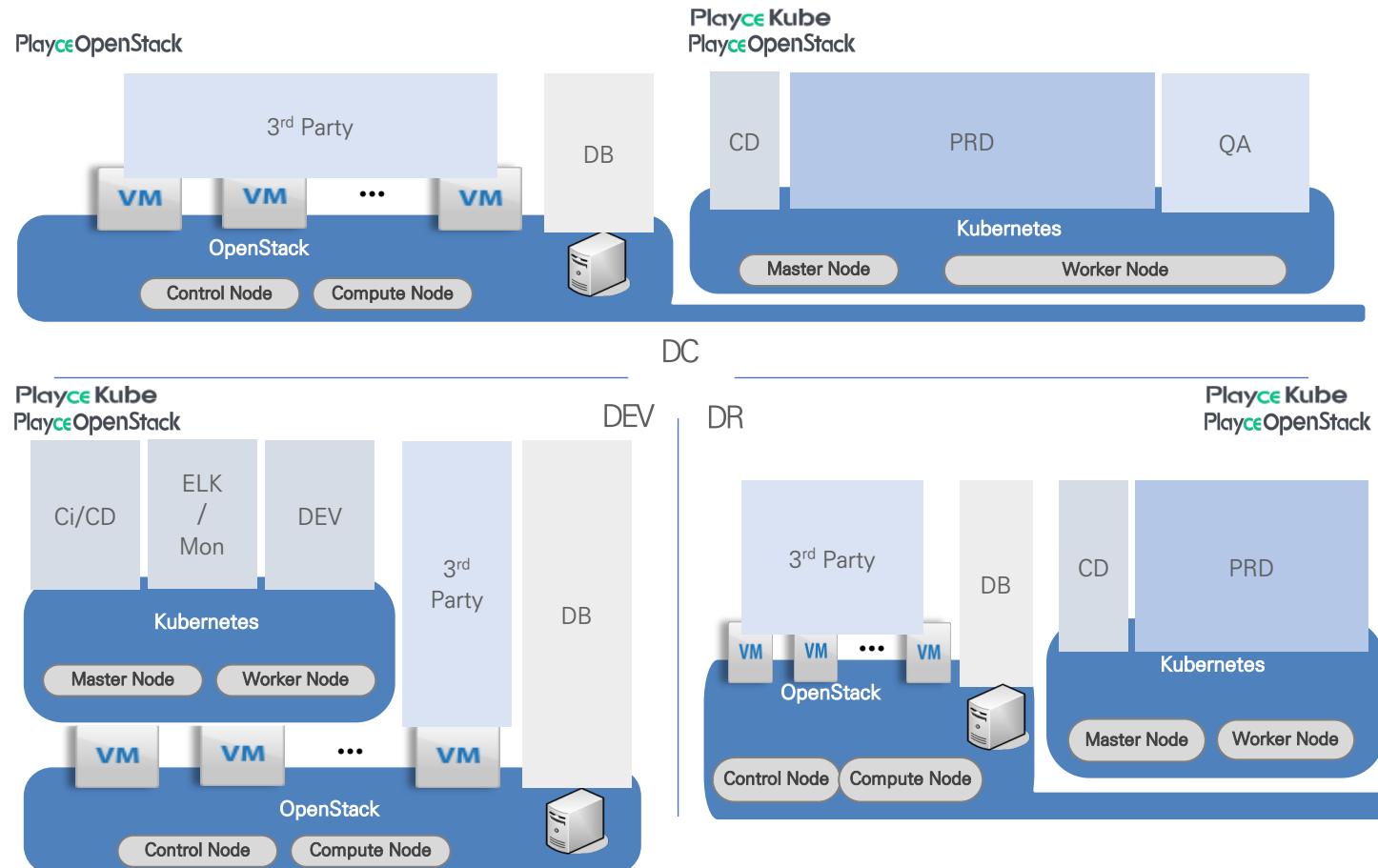
- 1000km 이상의 센터간 백업 가능한 클러스터 구성



# 구축 상세 사례 - 다용도 클러스터 구성

License 및 서비스 요구 조건에 따른 다양한 인프라 구성

## 서비스 구분별 클러스터 구성 아키텍처



## 아키텍처 특징 / 장점

OpenStack을 활용한 개발환경 운영환경 구성  
 필요시 DC-DR의 동시운영 가능성 확보

**DEV 환경의 자유로운 확장 운영**

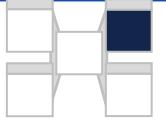
**3rd Party 제품 이중화 필수 (VM)**

**DC-DR 간 A-A 혹은 A-S 형태로  
운영 가능 (사용인구 2억명 이상)**

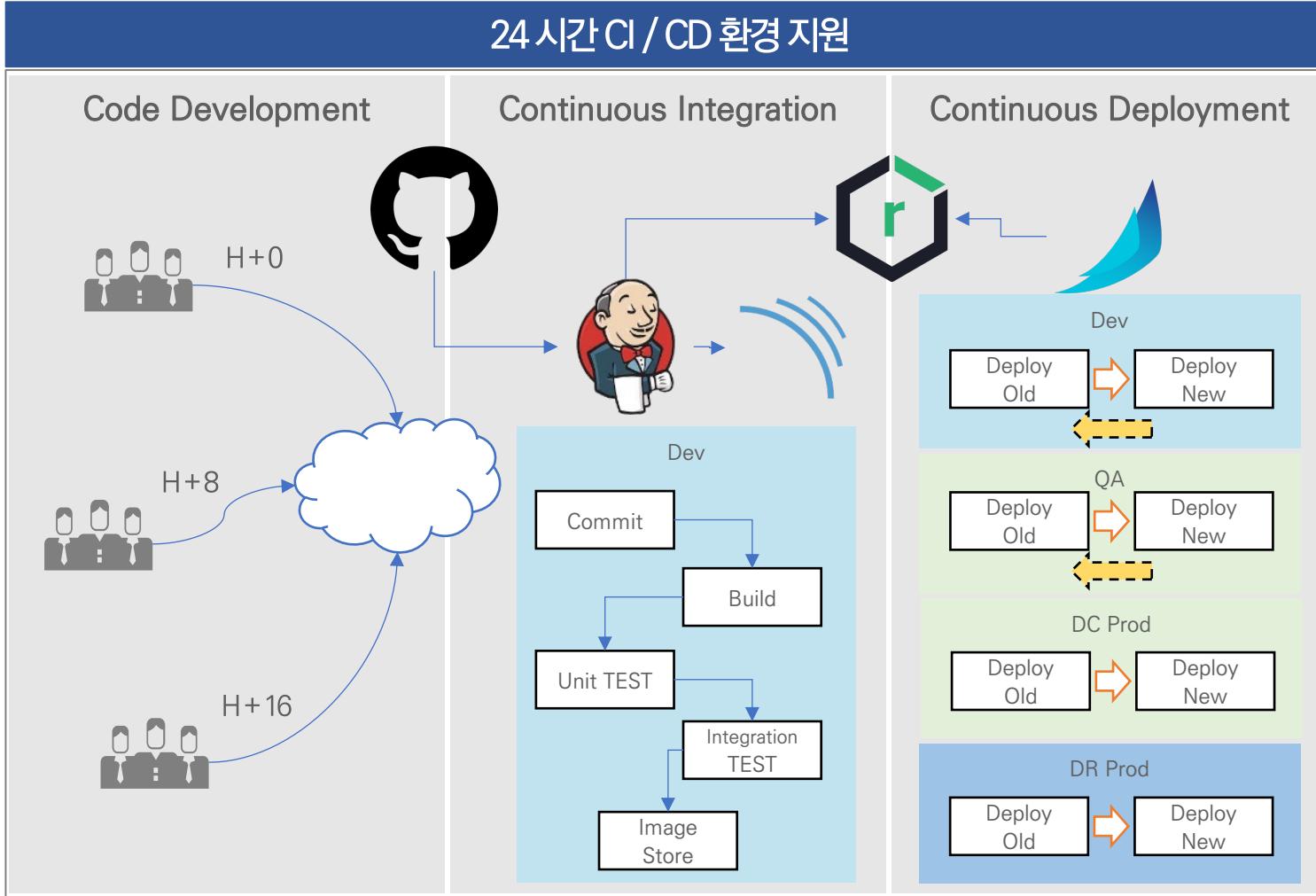
**GSLB를 통한 DC-DR 통제**

**3개국(다 지역) 동시 빌드 및  
지속 배포 지원**

# 구축 상세 사례 - CI/CD 구성



24시간 Build 및 CI / CD, 보안 및 테스트 지원



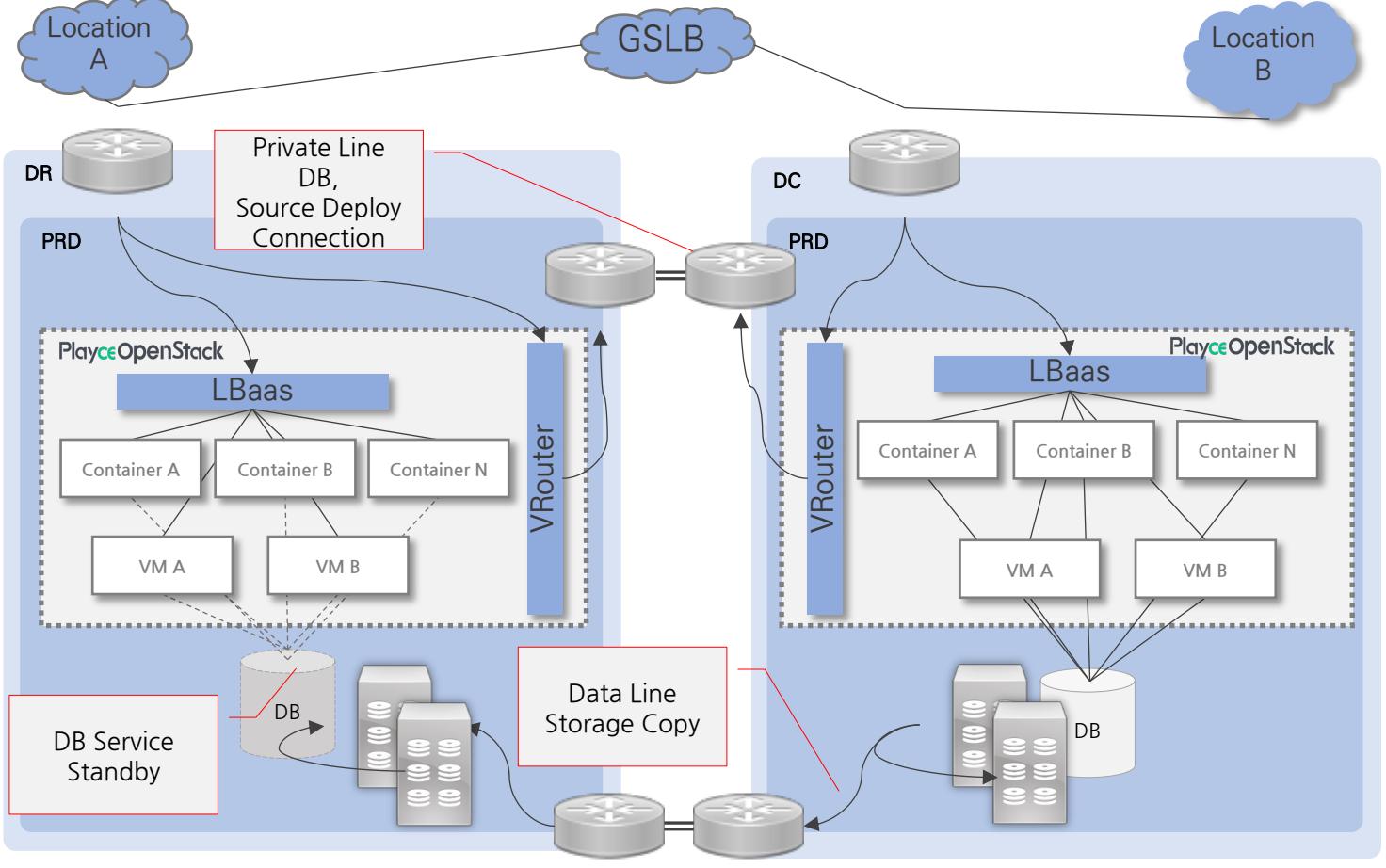
- ### 아키텍처 특징 / 장점
- 글로벌 개발을 위한 8시간 단위 Auto Commit – Build  
- Deploy 지원 필요  
DC 배포 실패를 방지하기 위한 QA, DC 동일 환경 생성 (Stage)
  - 8시간 단위  
Auto Commit – Build – Test
  - Kubernetes, VM 동시 배포 가능 환경
  - MSA를 적용한 배포 구조
  - Golden Container Image 관리

# 구축 상세 사례 - DR 클러스터 구성



OpenStack 을 이용한 환경과 Kubernetes 를 구분한 장애 대응 센터 구성

## DC – DR 환경의 Active – Active, Active – Standby 활용



## 아키텍처 특징 / 장점

2억명 이상의 사용자 대비와 빈번한 자연재해 대비를 위한 다용도, 다각도 아키텍처 구성

Storage 복제를 통한 DR 구성

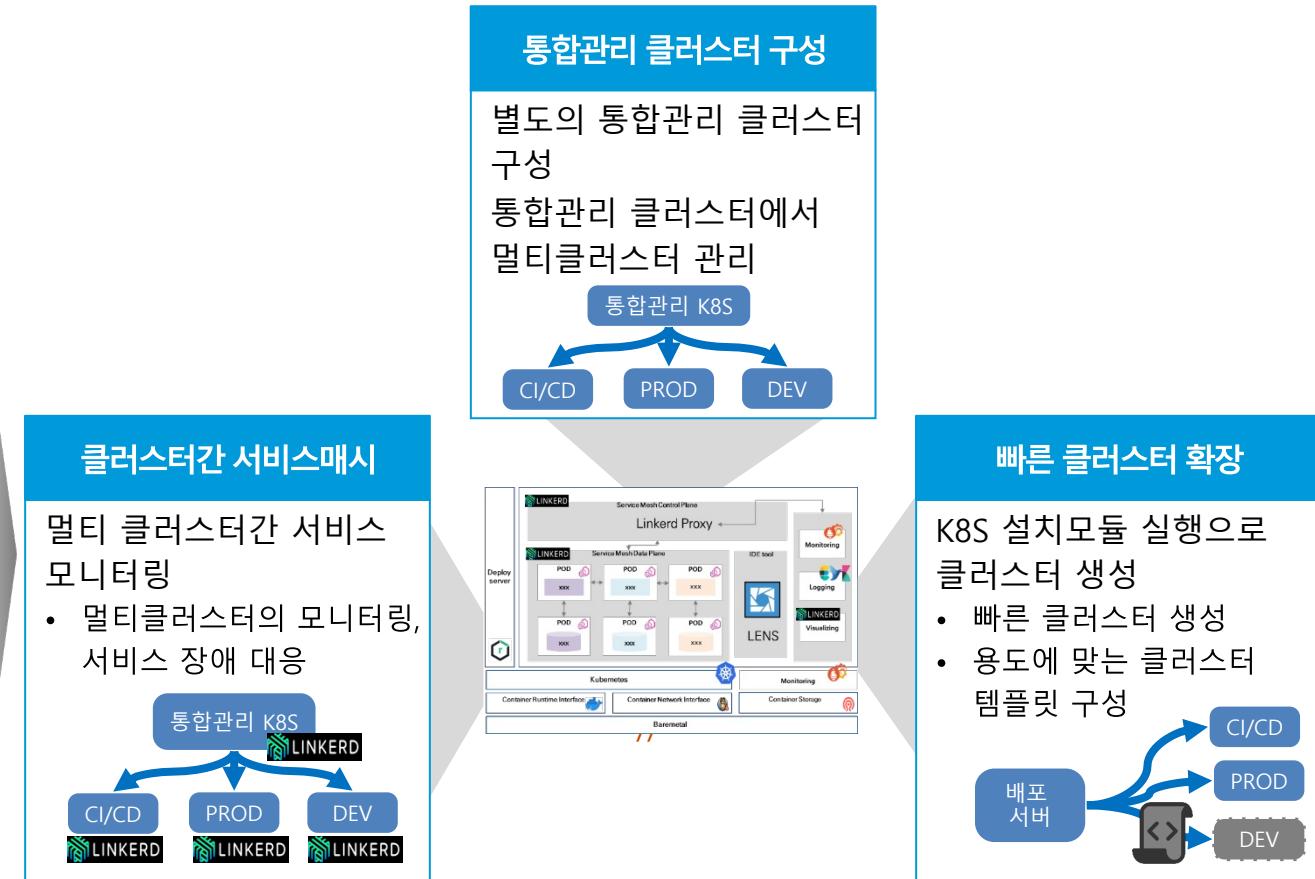
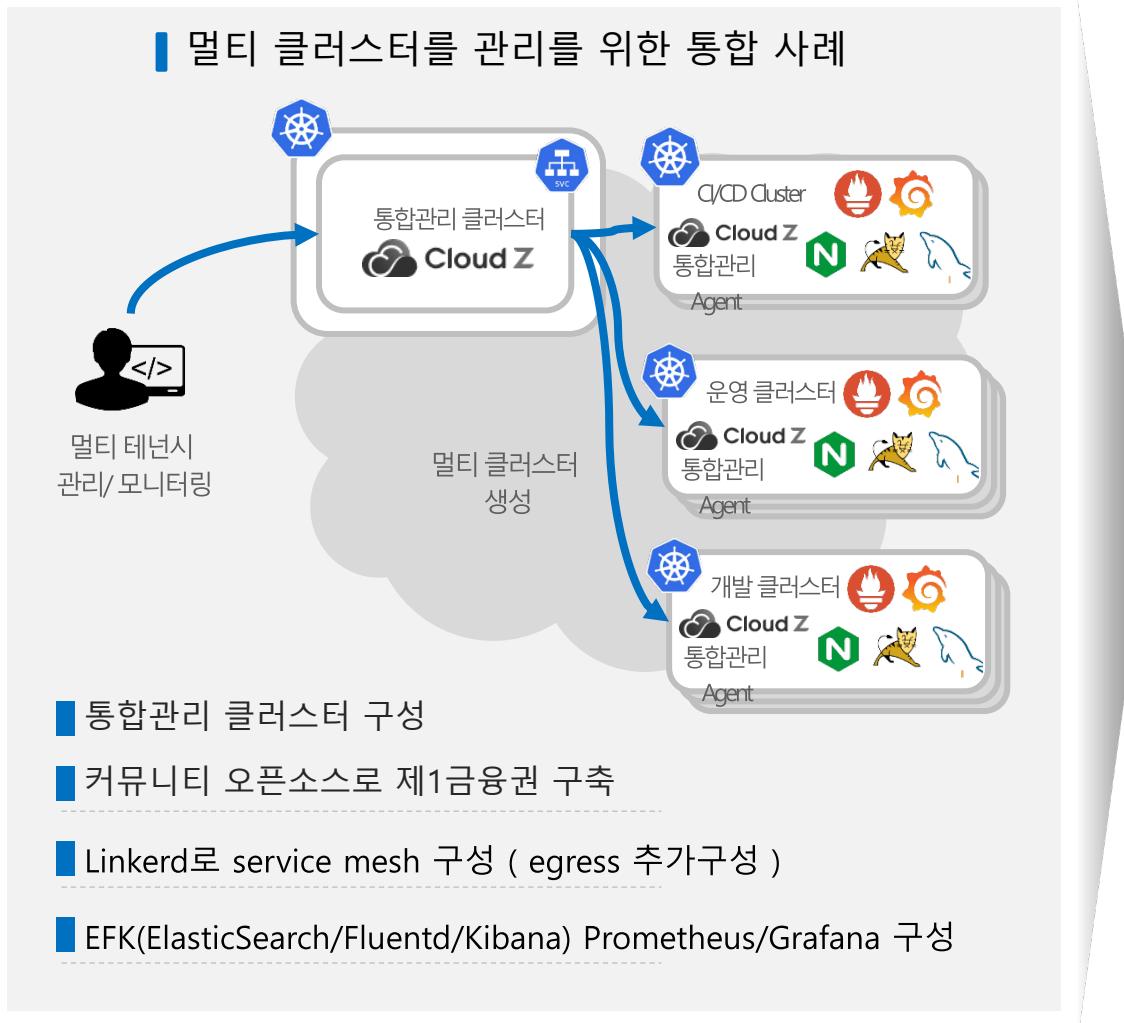
OpenStack 복제는  
Data volume 복제 구성

Kubernetes 복제는 NAS 복제 구성

Active – Active 구성을 대비한  
DB Data 구성

# 멀티 클러스터 통합 관리 구축 - XX 금융지주

멀티 클러스터 통합 관리 (운영, 모니터링, 로깅) 구성하고 필요시 즉각적으로 클러스터 확장 가능한 아키텍처 구성

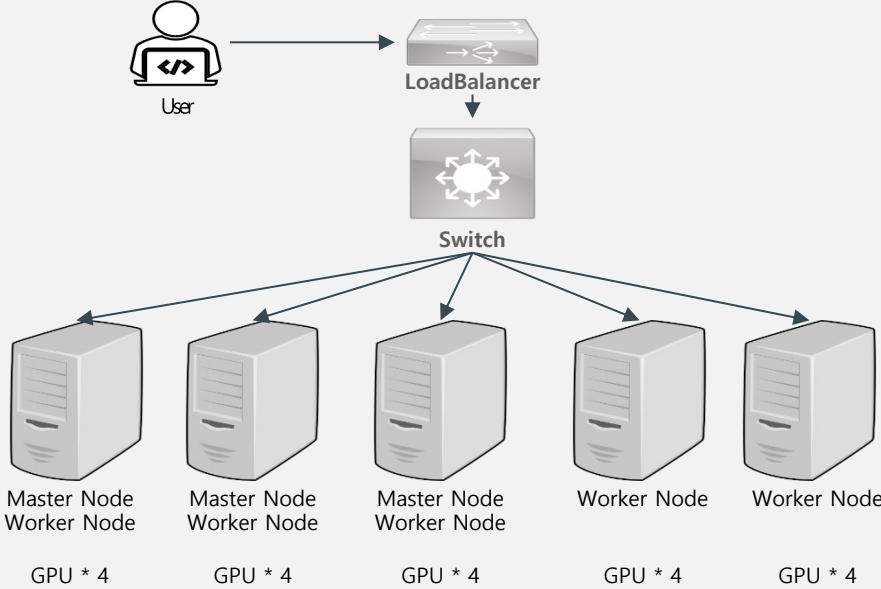


프로젝트 기간 : 2019.10 ~ 2019.12 / 2020.8~2020.12 / 2021~  
PaaS만 필요한 부분 / IaaS만 필요한 부분 / IaaS와 PaaS가 필요한 부분 / 개발자를 위한 PaaS 영역

# 독립적인 GPU Farm - X카드 AI 가상상담

- ML 학습 및 운영을 위해서 BareMetal 환경의 독립적인 K8S GPU Farm 환경 구성

## S사 AI 가상상담

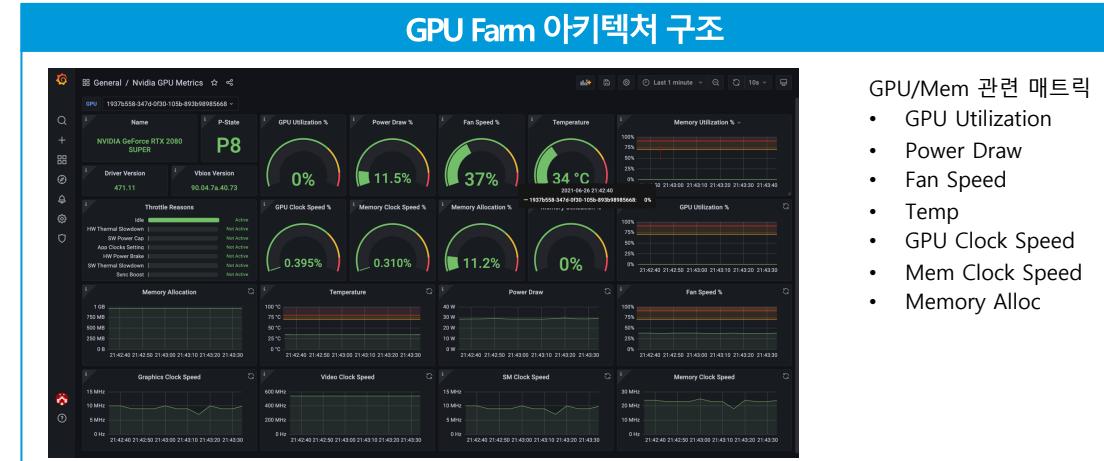
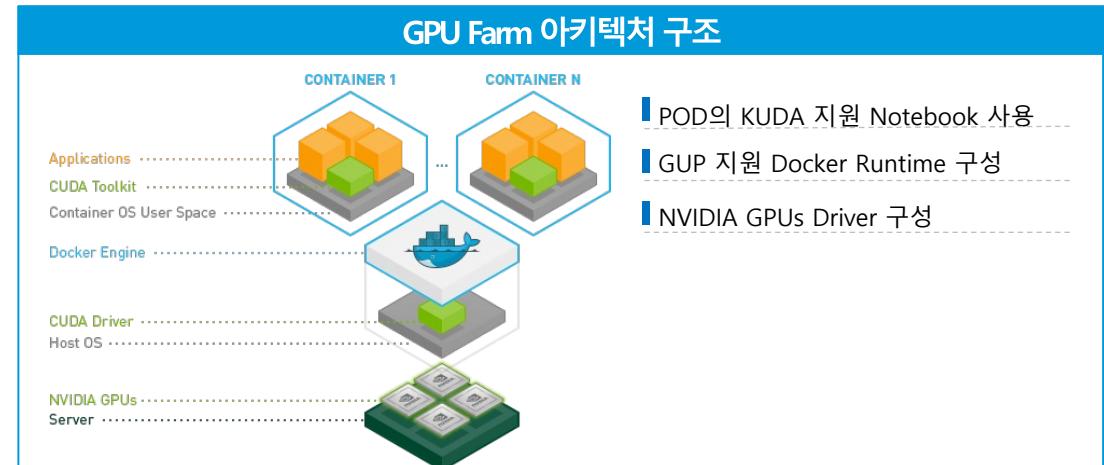


Kubernetes를 GPU 드라이버와 연동한 AI 가상상담 플랫폼 구축

Nvidia-Docker 런타임을 활용한 Kubernetes 구축

AI/ML 기반으로 작동하는 서비스를 대상으로 효율적인 스케줄링

Prometheus & Grafana를 통한 GPU 사용량 및 Metric 모니터링



프로젝트 기간 : 2021.6~ 2021.8  
PaaS 구축 AI 가상상담 플랫폼 운영환경 구축

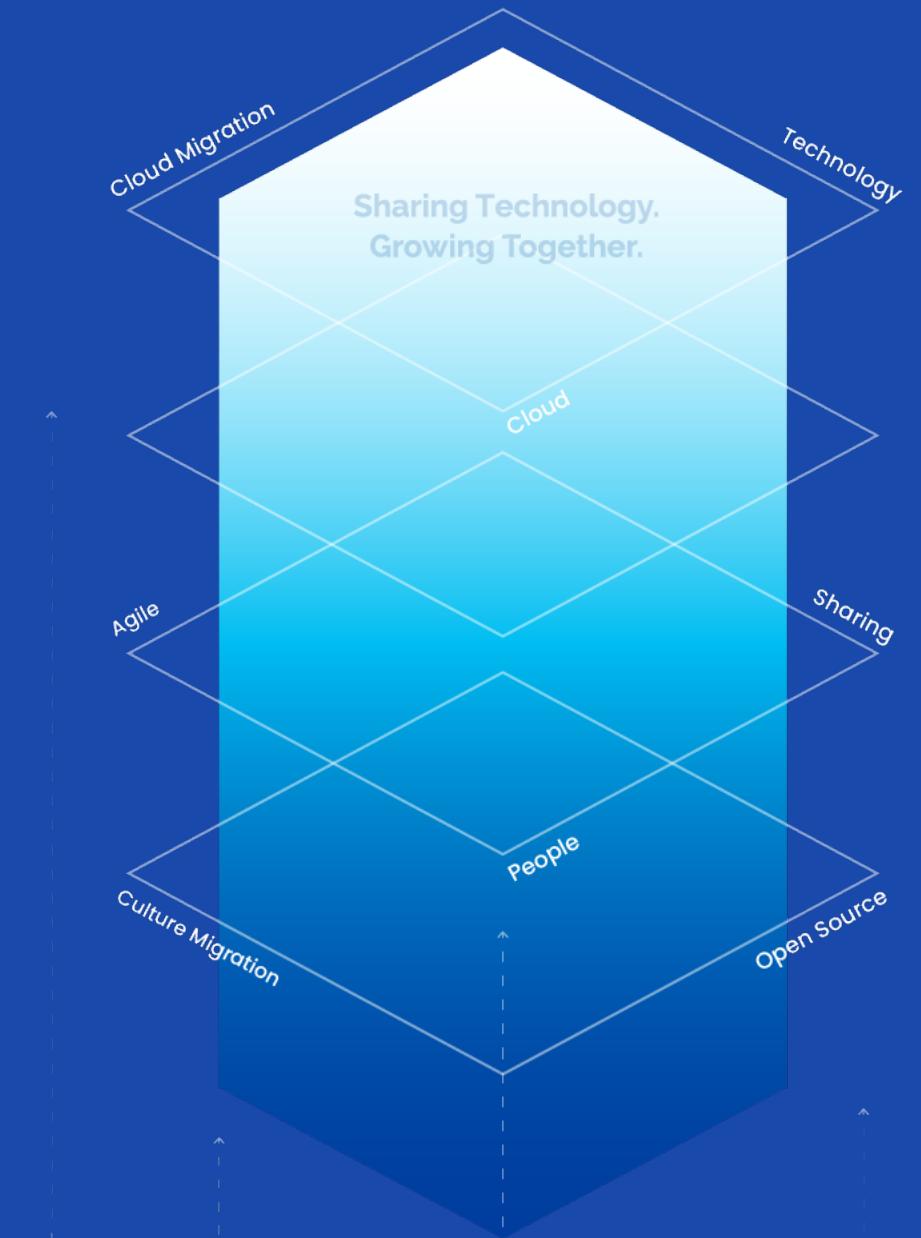
GPU/Mem 관련 매트릭

- GPU Utilization
- Power Draw
- Fan Speed
- Temp
- GPU Clock Speed
- Mem Clock Speed
- Memory Alloc

# Q&A

---

[sales@osci.kr](mailto:sales@osci.kr)



# Thank you.

감사합니다

[www.osci.kr](http://www.osci.kr)

T. 02-516-0711 E. [sales@osci.kr](mailto:sales@osci.kr)

